



Írta:

**PALÁGYI KÁLMÁN**

# **KÉPFELDOLGOZÁS HALADÓKNAK**

**Egyetemi tananyag**



**2011**



COPYRIGHT: © 2011–2016, Dr. Palágyi Kálmán, Szegedi Tudományegyetem  
Természettudományi és Informatikai Kar Képfeldolgozás és Számítógépes Grafika Tanszék

LEKTORÁLTA: Dr. Hajdu András, Debreceni Egyetem Informatikai Kar Komputergrafika  
és Képfeldolgozás Tanszék

Creative Commons NonCommercial-NoDerivs 3.0 (CC BY-NC-ND 3.0)

A szerző nevének feltüntetése mellett nem kereskedelmi céllal szabadon másolható,  
terjeszthető, megjelentethető és előadható, de nem módosítható.

#### TÁMOGATÁS:

Készült a TÁMOP-4.1.2-08/1/A-2009-0008 számú „Tananyagfejlesztés mérnök  
informatikus, programtervező informatikus és gazdaságinformatikus képzésekhez” című  
projekt keretében.



ISBN 978-963-279-518-8

KÉSZÜLT: a **Typotex Kiadó** gondozásában

FELELŐS VEZETŐ: Votisky Zsuzsa

AZ ELEKTRONIKUS KIADÁST ELŐKÉSZÍTETTE: Juhász Lehel

#### KULCSSZAVAK:

digitális képfeldolgozás, színes képek, matematikai morfológia, vázkijelölés,  
képszegmentáció, textúra-elemzés, kép-leképezések, képregisztráció, mozgóképek,  
többfelbontású képfeldolgozás.

#### ÖSSZEFOGLALÁS:

Ez a jegyzet a Szegedi Tudományegyetemen oktatott Képfeldolgozás haladóknak című  
tárgyhoz készült. A mesterszakos kurzus az alapozó Digitális képfeldolgozás tárgyra  
támaszkodva egyrészt a számítógépes képfeldolgozás további érdekes fejezeteibe nyújt  
betekintést, másrészt pedig az alapfejezetek újabb és fejlettebb technikáira tér ki. A jegyzet  
tíz fejezetből áll, melyek az alábbi területeket érintik: bináris tomográfia, halftoning,  
objektum-címkézés, a színes képek feldolgozása, matematikai morfológia, vázkijelölés,  
fejlett szegmentációs módszerek, textúra-elemzés, leképezések, warping, morphing,  
képregisztráció, mozgóképek feldolgozása, többfelbontású feldolgozás.

# Tartalomjegyzék

<b>Előszó</b>	<b>5</b>
<b>1. A bináris képekhez kapcsolódó területek</b>	<b>7</b>
1.1. Bevezetés a bináris tomográfiába . . . . .	7
1.2. Halftoning . . . . .	10
1.3. Objektumcímkézés . . . . .	14
<b>2. A színes képek és feldolgozásuk</b>	<b>17</b>
2.1. Színmodellek . . . . .	17
2.2. A színes képek feldolgozása . . . . .	21
<b>3. Matematikai morfológia</b>	<b>28</b>
3.1. Alapműveletek: dilatació és erózió . . . . .	28
3.2. Nyitás, zárás, morfológiai szűrés . . . . .	31
3.3. Hit-or-miss transzformáció . . . . .	34
3.4. Morfológiai vékonyítás . . . . .	35
3.5. Morfológiai váz . . . . .	35
3.6. Morfológia többszintű képekre . . . . .	37
<b>4. Vázkijelölés</b>	<b>42</b>
4.1. A váz meghatározása és tulajdonságai . . . . .	42
4.2. Távolság-alapú vázkijelölés . . . . .	51
4.3. A Voronoi váz . . . . .	57
4.4. A vékonyítás . . . . .	62
<b>5. Fejlett szegmentációs módszerek</b>	<b>67</b>
5.1. Szegmentálás deformálható határokkal . . . . .	67
5.1.1. Aktív kontúr . . . . .	67
5.1.2. Level set . . . . .	69
5.2. Vízválasztó szegmentálás . . . . .	70
5.3. Szegmentálás chamfer illesztéssel . . . . .	72
5.4. Szegmentálás fuzzy megközelítéssel . . . . .	74

<b>6. Textúra-elemzés</b>	<b>80</b>
6.1. Statisztikai textúra-leírás	81
6.1.1. Frekvencián alapuló módszerek	81
6.1.2. Együttes előfordulási mátrixok	83
6.2. Szintaktikus textúra-leírás	89
<b>7. Leképezések, warping, morphing</b>	<b>90</b>
7.1. Képek geometriai transzformációi	90
7.2. Warping TPS interpolációval	99
7.3. Morphing	102
<b>8. Képregisztráció</b>	<b>106</b>
8.1. A regisztráció és módszerek	106
8.2. Pontpárokon alapuló módszerek	111
8.2.1. Merev-test transzformáció keresése	113
8.2.2. Affin transzformáció keresése	114
8.2.3. Perspektív transzformáció	116
8.2.4. Polinomiális nemlineáris transzformációk	116
8.3. Kölesönös információn alapuló regisztráció	118
<b>9. Mozgóképek feldolgozása</b>	<b>120</b>
9.1. Különbbségképeken alapuló mozgás-analízis	122
9.2. Optikai áramlás	124
9.3. Mozgás-elemzés pont-megfeleltetéssel	125
<b>10. A többfelbontású feldolgozás és a wavelet-ek</b>	<b>127</b>
10.1. Kép-piramisok	127
10.2. A Haar transzformáció	130
10.3. Wavelet függvények	133
10.4. Wavelet transzformáció	137
<b>Irodalomjegyzék</b>	<b>143</b>

# Előszó

Ez a jegyzet a Szegedi Tudományegyetem (SZTE) programtervező informatikus MSc szakán a *Képfeldolgozás haladóknak* című tárgyhoz készült. Az SZTE Természettudományi és Informatikai Karán az Informatikai Tanszékcsoport által gondozott Mérnök informatikus BSc és a Programtervező informatikus BSc szakokon törzstárgy a *Digitális Képfeldolgozás*, melynek tematikája a tudományág legfontosabb területeit és annak alapvető technikáit, algoritmusait öleli fel. A mesterszakos Képfeldolgozás haladóknak az alapozó Digitális Képfeldolgozás tárgyra támaszkodva egyrészt a számítógépes képfeldolgozás további érdekes fejezeteibe nyújt betekintést, másrészt pedig az alapfejezetek újabb és fejlettebb technikáira tér ki.

A jegyzet 1. fejezete a bináris képekhez kötődő három területtel foglalkozik. Ezek közül az első a *bináris tomográfia*, ahol a bináris mátrixok kétirányú vetületekből (a sor- és az oszlop-összegekből) történő rekonstrukciójával foglalkozunk és bemutatjuk Ryser klasszikus algoritmusát. A bináris képekkel kapcsolatos második terület a *halftoning*, *dithering*, vagyis a többszintű képek binárisokkal történő helyettesítése. A többszintű és a színes képek nyomtatásakor kulcsfontosságú probléma megoldására a klasszikus pásztázó algoritmust mutatjuk be. A harmadik téma az *objektum-címkézés*, vagyis bináris képeken az objektumpontok összefüggő komponenseinek megcímkézése, mely pl. az objektum-számlálás és az objektumok statisztikai elemzése feladatokat megoldását segíti.

A 2. fejezetben a *színes képek* feldolgozását tárgyaljuk. A különböző színmodellek és a feldolgozási módok bemutatása után a képfeldolgozás számos területén (pl. kvantálás, pontoperációk, szűrés, simítás, éldetektálás, szegmentálás, képkódolás) a színes képekre javasolt módszereket tárgyaljuk.

A 3. fejezet bevezetés a *matematikai morfológia* eszköztárába. Bemutatjuk a pontthalmazokra vagy a bináris képekre az alpműveleteket és a belőlük építkező összetett operátorokat, majd kitérünk a többszintű képek morfológiai műveleteire is.

A 4. fejezetben a *váz* mint régió-alapú alakjellemzővel foglalkozunk. A váz meghatározására adott módszerek mellett további, a képfeldolgozás számos területén alkalmazható algoritmust (pl. távolságtranszformáció számítására, Voronoi diagramok meghatározására és a Delaunay felbontásra) is bemutatunk.

Az 5. fejezetben a fejlett szegmentációs módszerek négy csoportját tárgyaljuk: a deformálható határokon alapulókat (aktív kontúr, level set), a vízválasztó szegmentálást, a chamfer illesztést és a fuzzy-összefüggőségen alapulót.

A 6. fejezetet a *textúra-elemzés* módszereinek szenteltük. Bemutatjuk a statisztikai módszereket (a Fourier textúra függvényeken és az együttes előfordulási mátrixokon alapulókat) és érintjük a szintaktikus textúra leírást is.

A 7. fejezetben a *leképezésekkel* (vagyis képek geometriai transzformációinak végrehajtása) foglalkozunk, majd a *warping* és *morphing* (vagyis képek animált, nemlineáris átalakításainak) módszereit tekintjük át.

A *képregisztráció*nak (vagyis képek közötti geometriai viszony, transzformáció megkeresésének) és a hozzá kapcsolódó problémáknak szenteljük a 8. fejezetet, melyben áttekintjük a regisztráció, az arra épülő képpillesztés és képfűző feladatát, valamint a regisztrációs probléma megoldására javasolt módszereket. A terület feldolgozásánál részletesebben ismertetjük kontrollpont-párokon alapuló technikákat és a kölcsönös információt alkalmazó módszereket.

A 9. fejezetben a *mozgás*, a mozgás-követés és a mozgóképek feldolgozása kerül teríték-re. A különbség-képeken alapuló mozgás analízis mellett foglalkozunk mozgás-mező optikai áramlással történő közelítésével és az egymásnak megfeleltetett pontpárokon alapuló mozgáselemzéssel, mozgás-követéssel is.

Végezetül a 10. fejezetben az ugyancsak számos területen alkalmazható a *többsfelbontású feldolgozás*ba nyújtunk betekintést. A kép-piramisok és a Haar transzformáció ismertetése után a *wavelet*-eknek és wavelet transzformációnak a többsfelbontású feldolgozásban betöltött szerepét mutatjuk be.

Sajnos a digitális képfeldolgozás magyar nyelvű irodalma felettébb szűkös, így számos szakkifejezésnek még nincs konszenzusos magyar megfelelője. Így a jegyzetben a jól-rosszul honosított alapfogalmak mellett zárójelben feltüntettük az eredeti angol elnevezéseket is.

Hálás vagyok Dr. Balázs Péternek a bináris tomográfiát tárgyaló 1.1. alfejezet és Dr. Tanács Attilának a képregisztrációról szóló 8. fejezet elkészítésében nyújtott segítségükért. Köszönetemet fejezem ki valamennyi hallgatónak, aki a jegyzetben tárgyalt eljárások implementálásával és azok illusztrálásával segített. Köszönöm Dr. Hajdu Andrásnak, a jegyzet lektorának hasznos tanácsait.

A jegyzet a TÁMOP-4.1.2-08/1/A-2009-0008 (Tananyagfejlesztés mérnök informatikus, programtervező informatikus és gazdaságinformatikus képzésekhez) pályázat támogatásával jöhetett létre. Ezúton mondok köszönetet a támogatott pályázat elkészítőinek és a projekt irányítóinak is.

Dr. Palágyi Kálmán  
Szeged, 2011. június

# 1. fejezet

## A bináris képekhez kapcsolódó területek

A jelen fejezetben a bináris (kétszintű) képekhez kötődő három területtel foglalkozunk.

Az 1.1. alfejezetben a *bináris tomográfia* egyik problémájával, a bináris mátrixok kétirányú vetületekből (sor- és oszlop-összegekből) történő rekonstrukciójával foglalkozunk és bemutatjuk Ryser-algoritmusát. A *halftoning*, *dithering*, vagyis a többszintű képek binárisokkal történő helyettesítésének módszereibe nyújtunk betekintést az 1.2. alfejezetben. Az 1.3. alfejezetben az *objektum-címkézésre*, vagyis bináris képeken az objektumpontok összefüggő komponenseinek meghatározására mutatjuk be a klasszikus algoritmust.

### 1.1. Bevezetés a bináris tomográfiába

A diszkrét tomográfia speciális esete a *bináris tomográfia*, amikor a rekonstruálandó kép bináris, azaz csak fekete és fehér pixeleket tartalmaz. Ekkor az  $A = [a_{ij}]$   $m \times n$ -es képtömb a *horizontális és vertikális vetületei* (vagy *sor- és oszlopösszegei*), az  $R(A) = (r_1, \dots, r_m)$  és a  $S(A) = (s_1, \dots, s_n)$  vektorok ismeretében határozandó meg, ahol

$$r_i = \sum_{j=1}^n a_{ij} \quad (i = 1, \dots, m)$$

és

$$s_j = \sum_{i=1}^m a_{ij} \quad (j = 1, \dots, n).$$

A fenti probléma kapcsán felmerül a megoldhatóság (egzisztencia) és az egyértelműség (unicitás) kérdése. A megoldás létezéséhez nyilvánvalóan szükséges az, hogy a vetületekre teljesüljenek az alábbi feltételek:

- $\sum_{i=1}^m r_i = \sum_{j=1}^n s_j = M$ ,
- $r_i \leq n$  ( $i = 1, \dots, m$ ),
- $s_j \leq m$  ( $j = 1, \dots, n$ ).

A bináris tomográfia problémája megoldására Ryser 1957-ben közölte a következő klasszikus eljárást [54], mely feltételezi, hogy a vetületekre teljesülnek a fenti feltételek.

**Algoritmus:** Ryser

1. Vegyük az oszlopösszegek  $S$  vektora komponenseinek egy nemnövekvő  $\pi$  permutációját és jelölje  $S'$  a permutált vektort.
2. Konstruáljuk meg az  $R$  sorösszegek alapján balról feltöltött megfelelő  $A^* = [a^*_{ij}] m \times n$ -es mátrixot, ahol:

$$a^*_{ij} = \begin{cases} 1, & \text{ha } j \leq r_i \\ 0, & \text{különben} \end{cases}$$

és jelölje  $S^* = (s^*_1, \dots, s^*_n)$  az  $A^*$  mátrix oszlopösszegeinek vektorát.

3. A  $k = n, \dots, 2$  oszlopindexekre rendre hajtjuk végre a következőt: ha  $s^*_k < s'_k$ , akkor az  $A^*$  első  $k-1$  oszlopából álló mátrix legjobboldali 1-eseket tartalmazó  $l$ -edik oszlopából toljunk át  $s^*_k - s'_k$  darab 1-et a  $k$ -adik oszlopba azokon a sorokon belül, ahol az  $l$ -edik oszlopban 1-es, a  $k$ -adikban pedig 0 áll. Amennyiben választhatunk, akkor a legfelsőbb sorokban lévő 1-eseket toljuk át.
4. Rendezzük az  $A^*$  mátrix oszlopokat a  $\pi^{-1}$  permutáció segítségével.

Az algoritmust az alábbi példán szemléltetjük. Legyen  $R=(2,4,3,4,1)$  és  $S=(3,4,3,2,1,1)$ . Ekkor az algoritmus 1. lépése után  $S' = (4,3,3,2,1,1)$ , amit például a  $\pi = [2, 1, 3, 4, 5, 6]$  permutáció alkalmazásával kaptunk. A 2. lépésben inicializált mátrix

$$A^* = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

melyre  $S^*=(5,4,3,2,0,0)$ . Az algoritmus további lépései során a következő mátrixokat kapjuk (minden esetben csillaggal jelöljük azokat az elemeket, amelyeket jobbra mozgatunk, és  $\circ$  jellel azt, ahova mozgatjuk őket az adott sorban).

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1^* & 0 & 0^\circ \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1^* & 0^\circ & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1^* & 0^\circ & 0 & 1 \\ 1 & 1 & 1^* & 0^\circ & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1^* & 0^\circ & 0 & 0 & 0 \\ 1 & 1^* & 0^\circ & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1^* & 0^\circ & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Végezetül a kapott mátrix első és második oszlopának felcserélésével kapjuk az eredeti feladat megoldását:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Mivel a 2. lépésben inicializált  $A^*$  mátrix minden sora pontosan az előírt sorösszeggel megegyező számú 1-est tartalmaz, így kezdetben a vízszintes vetületek kielégítettek. Ez a tulajdonság az 1-esek jobbratolásával a 3. lépés során és az oszlopok permutációjakor (a 4. lépésben) is teljesül, így a létrehozott mátrix sorait tekintve korrekt a rekonstrukció eredménye.

Nézzük az oszlopösszegek alakulását. Tudjuk, hogy fennáll a  $\sum_{i=1}^m r_i = \sum_{j=1}^n s_j = M$  egyenlőség, vagyis a rekonstruálandó mátrix pontosan  $M$  darab 1-est tartalmaz. Ebből adódik, hogy az oszlopokban az eljárás 3. lépésében a hiányzó 1-esek száma megegyezik a többlet egyesek számával. Az 1-eseket a nemnövekvő átrendezés miatt mindig elegendő balról jobbra tolni. Tehát a jobbratologatások során az 1-esek hiányai pontosan kiegyenlíthetők, és így az algoritmus terminálásakor az oszlopösszegek is korrektek lesznek.

Az  $S$  oszlopösszeg vektor nemnövekvő átrendezése (egy optimális komplexitású rendezési algoritmus alkalmazásával)  $O(n \log n)$  időt vesz igénybe. Az  $A^*$  mátrix inicializálása és a legfeljebb  $O(mn)$  darab 1-es jobbratologatása  $O(mn)$  idő alatt elvégezhető, így az algoritmus futási ideje legrosszabb esetben  $m \times n$ -es mátrixokra  $O(mn + n \log n)$ .

A példaként bemutatott rekonstrukciós feladat megoldása nem egyértelmű, mivel előfordult az, hogy többféleképpen is kiválaszthatunk volna az adott oszlopból a jobbratolandó 1-eseket (nem csak úgy, hogy a legfelsőbb sorokban lévőket mozgattuk meg). A már bemutatott megoldáson kívül pl. az alábbi is megkaphattuk volna:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

A bináris tomográfia fenti problémájának megoldása akkor és csakis akkor egyértelmű, ha (pl. a Ryser-algoritmussal) megtalált  $A^*$  megoldásnál nincs olyan  $(i_1, j_1)$  és  $(i_2, j_2)$  indexpár, amelyre

$$a^*_{i_1, j_1} = a^*_{i_2, j_2} = 0$$

és

$$a^*_{i_1, j_2} = a^*_{i_2, j_1} = 1,$$

vagyis nem találunk a kapott megoldásban olyan téglalapot, amelynél az egyik átló szerinti sarkok 0-, a másik átló végpontjai pedig 1-értékűek. Az ilyen tulajdonságú téglalapokat *kapcsoló komponenseknek* nevezik. Az unicitásra vonatkozó ekvivalencia-tétel egyik iránya (mivel szerint több megoldás létezik, ha szerepel az  $A^*$ -ben kapcsoló komponens) triviális, hiszen



nyilvánvaló, hogy a kapcsoló komponens átlóinak átszínezésével a vetületek változatlanok maradnak, vagyis egy újabb megoldáshoz jutunk.

A jelen jegyzet kereteit sajnos szétfeszítené, ha a diszkrét tomográfia területének még további érdekes problémáira és azok megoldásaira is kitérnénk. A téma iránt érdeklődők figyelmébe a Gábor T. Herman és Kuba Attila szerkesztette könyveket [30, 31] és Balázs Péter jegyzetét [4] ajánljuk.

## 1.2. Halftoning

A többszintű (szürkeárnyaltos) és a színes képek nyomtatásakor egyaránt azt kell megadni, hogy a festékkazettából a nyomtatandó felület mely pozícióira jusson festék, vagyis (a nyomtatandó kép fizikai mérete és a nyomtató felbontása által meghatározott méretű) bináris képeket, az úgynevezett *féltonusú* (*halftone*) képeket kell előállítani. Színes képek nyomtatásakor a CMYK színtér komponenseihez tartozó négy bináris képpel helyettesítjük az eredetit. A továbbiakban a szürkeárnyaltos képek konverziójára szorítkozunk.

A kvantálási szintek számát drasztikusan csökkentő halftoning eljárások alapvetően két csoportba oszthatók:

- az előállított bináris kép mérete nagyobb a konvertálandó többszintű kép méreténél,
- az előállított bináris kép mérete megegyezik a konvertálandó többszintű kép méretével.

Az első technika alkalmazhatósága erősen korlátozott, hiszen mindössze  $(k^2+1)$ -féle  $k \times k$  méretű bináris mintát vehetünk be ( $k \geq 2$ ), melyek rendre  $0, 1, \dots, k^2$  darab fekete pontot tartalmaznak, így a képpontok számának  $k^2$ -szeres növelése árán a legfeljebb  $(k^2+1)$ -szintű képeket helyettesíthetünk binárisokkal. Az 1.1. ábrán a 10-féle  $3 \times 3$ -as mintát mutatjuk be. A mintával helyettesítés az alábbi általános sémát követi:

**Algoritmus:** helyettesítés mintával

*Input:* az  $A = [a(i, j)]$   $m \times n$ -es,  
legfeljebb  $(k^2+1)$ -féle intenzitást tartalmazó kép és  
az  $X^s = [x^s(p, q)]$   $k \times k$  méretű bináris képek ( $s = 0, 1, \dots, k^2$ )

*Output:* a  $B = [b(u, v)]$   $(k \cdot m) \times (k \cdot n)$ -es bináris kép

```
for  $i = 0$  to  $m - 1$  do
  for  $j = 0$  to  $n - 1$  do
    for  $p = 0$  to  $k - 1$  do
      for  $q = 0$  to  $k - 1$  do
         $b(k \cdot i + p, k \cdot j + q) = x^{a(i, j)}(p, q)$ 
```

A második megközelítésre, vagyis a mérettartó konverzióra számos módszert javasoltak.

A kvantálási hibák csökkentésére is alkalmazott *dithering* technikát a jelen probléma megoldására is javasolták: Előállítunk egy  $k \times k$ -méretű ( $k \geq 2$ ) dither mátrixszal „kicsempézett” dither-képet. Az output kép valamely pontja fekete/kitöltött, ha az input kép megfelelő pontjának intenzitása nagyobb a dither kép ugyanazon pozícióján álló elemnél. A módszer pszeudókódja:

**Algoritmus:** dithering

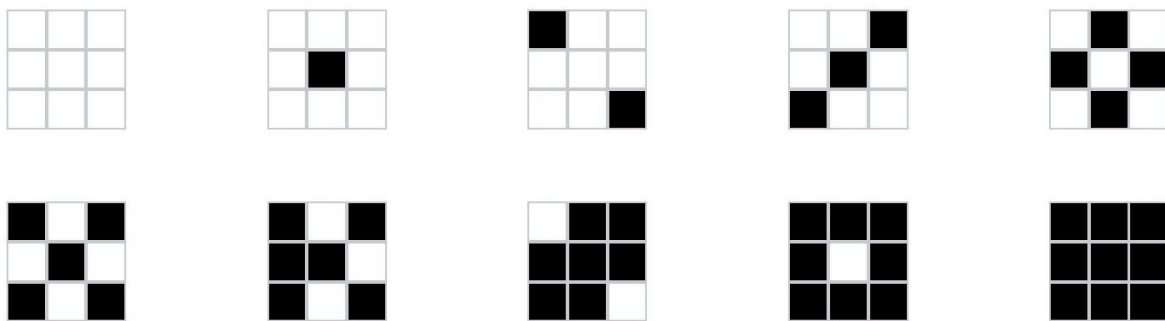
*Input:* az  $A = [a(i, j)]$   $m \times n$ -es kép és  
egy  $D = [d(p, q)]$   $k \times k$  méretű dither mátrix

*Output:* a  $B = [b(u, v)]$   $m \times n$ -es bináris kép

```

for  $i = 0$  to  $m - 1$  do
  for  $j = 0$  to  $n - 1$  do
    if  $a(i, j) > d(i \bmod k, j \bmod k)$  then
       $b(i, j) = 0$  // fekete
    else
       $b(i, j) = 1$  // fehér

```



1.1. ábra.  $3 \times 3$ -as minták, melyekkel a legfeljebb 10-szintű képek pontjai helyettesíthetők.

Előre adott dither mátrixok helyett az input kép pontjainak intenzitásait (pontonként generált) véletlen számokkal is össze szokták vetni: az eredménykép adott pontja fekete, ha az ugyanazon pozíciójú input pixel intenzitása nagyobb a hozzá generált számnál. A random dithering eljárásra az 1.2. ábrán mutatunk példát.

A dithering eljárásai leggyakrabban a *hiba-elmosás* (*error diffusion*) megközelítést követik. A módszerrel akár helyben, az input többszintű kép felülírásával is előállíthatjuk a bináris képet egyetlen szekvenciális (pl. sorfolytonos vagy serpentin) bejárással.

Ha a többszintű képen az éppen meglátogatott képpont intenzitása meghaladja a maximális intenzitás felét, akkor az output képpont értéke „fehér” lesz, ellenkező esetben pedig „fekete”. A módszer tehát azonnali és visszavonhatatlan döntést hoz a bináris kép aktuális pontja értékéről. A döntés több (az intenzitáskészlet felének megfelelő számú) értékhez rendel ugyanazt az outputot. A módszert az különbözteti meg az egyszerű küszöböléstől, hogy a „hibát” (vagyis a „fehérnek” minősített intenzitás esetében a bemenő intenzitás és a maximális intenzitás különbségét) súlyozottan szétteríti az aktuális képpont egy (még bejáratlan) lokális környezetében. A módszer pszeudokódja:



1.2. ábra. Példa random dithering-re. A 256 szürkeségi szinttel kvantált Lena kép (bal) és annak a random dithering technikával előállított bináris megfelelője (jobb).

#### Algoritmus: Floyd-Steinberg

*Input:* az  $A = [a(i, j)]$   $m \times n$ -es kép  $L$  intenzitással

*Output:* a  $B = [b(u, v)]$   $m \times n$ -es bináris kép

```

for  $i = 0$  to  $m - 1$  do
  for  $j = 0$  to  $n - 1$  do
    if  $a(i, j) < L/2$  then
       $b(i, j) = 0$  // fekete
    else
       $b(i, j) = 1$  // fehér
       $error = a(i, j) - b(i, j) * (L - 1)$ 
       $a(i, j + 1) = a(i, j + 1) + error * (7/16)$ 
       $a(i + 1, j - 1) = a(i + 1, j - 1) + error * (3/16)$ 
       $a(i + 1, j) = a(i + 1, j) + error * (5/16)$ 
       $a(i + 1, j + 1) = a(i + 1, j + 1) + error * (1/16)$ 

```

Floyd-Steinberg hiba-elmosó algoritmusát tehát a kvantálási hibát az aktuális képpont  $3 \times 3$ -as környezetének pontjai között az alábbi maszk szerinti súlyokkal teríti szét:

$$\frac{1}{16} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{pmatrix}.$$

Az eljárás hatását az 1.3. ábra illusztrálja.



1.3. ábra. A 256 szürkeségi szinttel kvantált Lena kép (bal) és annak a Floyd-Steinberg hibaelmosó dithering technikával előállított bináris megfelelője (jobb).

További súlymátrixokat is javasoltak, melyek közül ötöt mutatunk be. A Jarvis-Judice-Ninke szűrő  $5 \times 5$ -ös környezettel dolgozik:

$$\frac{1}{48} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{pmatrix}.$$

A Stucki szűrő mérete szintén  $5 \times 5$ -ös:

$$\frac{1}{42} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{pmatrix}.$$

A Burkes szűrő nem négyzetes, mérete  $3 \times 5$ -ös:

$$\frac{1}{32} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \end{pmatrix}.$$

Sierra az alábbi  $5 \times 5$ -ös és  $3 \times 5$ -ös szűrőket javasolta:

$$\frac{1}{32} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 3 \\ 2 & 4 & 5 & 4 & 2 \\ 0 & 2 & 3 & 2 & 0 \end{pmatrix},$$

$$\frac{1}{16} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 3 \\ 1 & 2 & 3 & 2 & 1 \end{pmatrix}.$$

A különböző hiba-elmosó módszerek implementációjakor kihasználható az, ha a súlyok számlálóját vagy nevezőjét kettőhatvány, hiszen ekkor a szorzás és az osztás a gyorsabb léptetőoperátorokkal helyettesíthető. A súlyok összege „1” (valamennyi súlyozás esetén).

A halftoning probléma gazdag irodalmából R. Ulichney áttekintő kézikönyvét [72] ajánljuk az olvasó figyelmébe.

### 1.3. Objektumcímkézés

Számos alkalmazásnál fontos a bináris (szegmentált) képek összefüggő objektumainak egyedi címkével való ellátása. A címkézés a bemeneti bináris kép objektumpontjainak (a figyelembbe vett szomszédsági reláció reflexív-tranzitív lezártjával kapott ekvivalencia-reláció szerinti) osztályozása. A 2-dimenziós négyzetmozaikon (vagy a vele duális négyzetrácson) mintavételezett képeken a két leggyakrabban alkalmazott szomszédsági reláció a 4- és a 8-szomszédság. Egy képpont 4-szomszédainak koordinátái a vizsgált ponttól legfeljebb egy komponensben térhetnek el  $\pm 1$ -gyel, míg a 8-szomszédok mindkét koordinátában különbözhetnek  $\pm 1$ -gyel. A címkézésre az 1.4. ábrán mutatunk példát.

<p><b>"If you would know what the Lord God thinks of money, you have only to look at those to whom he gives it."</b></p>	<p><b>"If you would know what the Lord God thinks of money, you have only to look at those to whom he gives it."</b></p>
--	--

1.4. ábra. Egy bináris kép (bal) és azon a 8-összefüggő objektumok címkézése (jobb). (A címkézett kép álszínezéssel megjelenített.)

A címkézési feladat megoldására adhatnánk nyers erővel is megoldást: pl. keressük meg (mondjuk sorfolytonosan) az első objektumpontot, rendeljük ahhoz az első címkét, adjuk tovább azt a címkét a címkézett ponttal szomszédos objektumpontoknak és folytassuk tovább a címke-terjesztést, amíg az első címkével megcímkézett pontoknak található címkézetlen objektumpont szomszédja. Ha már nincs több ilyen pont, akkor keressünk egy újabb címkézetlen pontot, kapja meg az a következő címkét és terjesszük azt tovább az első címkéhez hasonló módon. Az eljárás terminál, ha már nem találunk további címkézetlen pontot a képen.

A fenténél jóval hatékonyabb, ráadásul könnyebben is implementálható eljárásokat is javasoltak [60]. A klasszikus címkézési algoritmus:



**Algoritmus:** objektumcímkézés

*Input:* az  $A = [a(i, j)]$   $m \times n$ -es bináris kép

*Output:* a  $L = [l(i, j)]$   $m \times n$ -es címketömb

```
// inicializálás
for  $i = 0$  to  $m - 1$  do
    for  $j = 0$  to  $n - 1$  do
         $l(i, j) = 0$ 
     $Eq\_Table = \langle \rangle$ 
     $label = 0$ 
// 1. sorfolytonos pásztázás
for  $i = 0$  to  $m - 1$  do
    for  $j = 0$  to  $n - 1$  do
        if  $a(i, j) = 1$  then
             $N = Labels\_in\_Visited\_Neighbors(L, i, j)$ 
            if  $Is\_Empty(N)$  then
                 $label = label + 1$ 
                 $l(i, j) = label$ 
            else
                 $l(i, j) = Min\_Label(N)$ 
                if  $Is\_Not\_Singleton(N)$  then
                     $Add(Eq\_Table, A)$ 
// Ekvivalenciaosztályok megkeresése
 $Eq\_Class = Resolve(Eq\_Table)$ 
// 2. sorfolytonos pásztázás
for  $i = 0$  to  $m - 1$  do
    for  $j = 0$  to  $n - 1$  do
        if  $a(i, j) = 1$  then
             $l(i, j) = Eq\_Class(l(i, j))$ 
```

Az eljárás kezdetén inicializáljuk, lenullázzuk az  $L$  címketömböt, üres lista lesz a kezdeti  $Eq\_Table$  ekvivalencia-tábla és a  $label$  aktuális címkét 0-ra állítjuk.

Az input bináris kép sorfolytonos bejárásakor minden egyes objektumpont esetén megvizsgáljuk a címketömbben a már meglátogatott szomszédokat. Háromféle esetet különböztetünk meg:

1. Ha nincs a szomszédságban kiosztott címke, akkor inkrementáljuk a  $label$  aktuális címkét és azt az új címkét rendeljük az aktuális objektumponthoz.
2. Ha a megcímkézett szomszédok ugyanazon címkével bírnak, akkor az aktuális pont is ugyanazt a címkét kapja meg.
3. Ha a szomszédok között többféle címke fordul elő, akkor az aktuális pont címkéje az  $N$ -beli címkék minimuma lesz és feljegyezzük az  $Eq\_Table$  ekvivalencia-táblába azt, hogy a pontunk körül előforduló címkék valamennyien ugyanahhoz az összefüggő objektumhoz tartoznak. (Megjegyezzük, hogy a 4- és a 8-szomszédság esetén legfeljebb kettő különböző címkét oszthatunk ki a már meglátogatott szomszédok között.)

Az *Eq\_Table* ekvivalencia-tábla alapján az *Eq\_Class* ekvivalencia-osztályok egy egyszerű mélységi kereséssel megkaphatók, az osztályok reprezentáns elemeit pedig egy második pásztázással rendeljük hozzá az *L* címketömbhöz.

Az első pásztázás menetét az 1.5. ábrával szemléltetjük.



1.5. ábra. A bemenő bináris kép (bal) és a címketömb az első pásztázási menet után (jobb). Az *Eq\_Table* ekvivalencia-táblába az  $\{1,2\}$  és az  $\{1,3\}$  halmazok kerülnek be, melyekből az  $\{1,2,3\}$  ekvivalenciaosztályt kapjuk, tehát a képen egyetlen 8-összefüggő objektum található.

## 2. fejezet

# A színes képek és feldolgozásuk

A színek két szempont miatt is fontosak a digitális képfeldolgozásban: Egyrészt a színinformáció gyakran megkönnyíti az alakfelismerést, másrészt pedig az emberi szem színek ezreit képes egymástól megkülönböztetni, szemben a szürke néhány tucat árnyalatával.

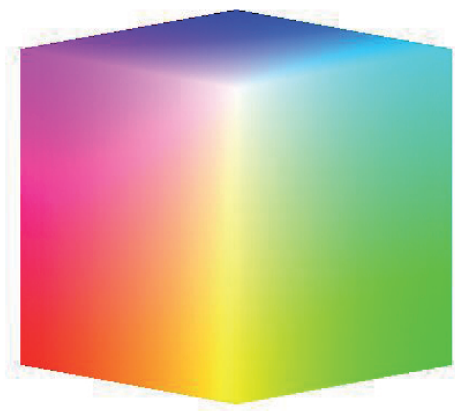
A jelen fejezet első pontjában bemutatjuk a leggyakrabban alkalmazott színmodelleket, a másodikban pedig áttekintjük a digitális képfeldolgozás néhány fontos területét színes képekre.

### 2.1. Színmodellek

A színes digitális képek ábrázolhatók akár skalár értékű képpontokkal is (ahol a képpontok intenzitása a hullámhossznak felel meg), de inkább 3- vagy 4-komponensű vektorokat rendelünk hozzájuk az alkalmazott *színmodell* szerint, ahol minden egyes színnek az adott színmodellhez tartozó *színtér* egy pontja felel meg.

A következőkben a leggyakrabban használt színmodelleket tekintjük át.

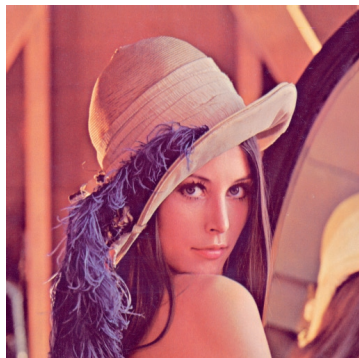
#### RGB



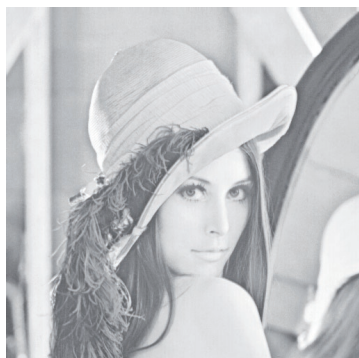
2.1. ábra. Az RGB színtere. A kocka takarásban lévő „fekete” sarka a koordináta-rendszer origója.



Az RGB (Red, Green, Blue) színmodellnél a három spektrális komponensnek a Descartes koordináta-rendszer tengelyei felelnek meg, szintere egy kocka-alakú tartomány. A normalizált RGB egységkockájának (ld. 2.1. ábra) nyolc csúcsa:  $(0,0,0)$  – fekete,  $(1,0,0)$  – vörös,  $(0,1,0)$  – zöld,  $(1,1,0)$  – sárga,  $(0,0,1)$  – kék,  $(1,0,1)$  – magenta,  $(0,1,1)$  – cián,  $(1,1,1)$  – fehér. A színtérben a szürkeárnyaltos skála megfelel a tartomány „fehér” és „fekete” csúcsait összekötő testátlónak (ahol megegyezik mindhárom színek komponens). Az RGB színmodell hardver-orientált, azt használják a színes monitorok és a színes kamerák.



RGB kép



R komponens



G komponens



B komponens

2.2. ábra. Egy színes RGB kép és komponensei monokróm képekként megjelenítve.

## CMY, CMYK

A CMY (Cyan, Magenta, Yellow) és a CMYK (Cyan, Magenta, Yellow, Key) színmodellek szintén hardver-orientáltak, a színes nyomtatók és másolók alkalmazzák őket. A normalizált  $RGB \rightarrow CMY \rightarrow C'M'Y'K$  konverziókat az alábbi formulák írják le:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

$$\begin{bmatrix} C' \\ M' \\ Y' \\ K \end{bmatrix} = \begin{bmatrix} (C - K)/(1 - K) \\ (M - K)/(1 - K) \\ (Y - K)/(1 - K) \\ \min\{C, M, Y\} \end{bmatrix}, \text{ ha } (C, M, Y) \neq (1, 1, 1) \text{ és}$$

$$\begin{bmatrix} C' \\ M' \\ Y' \\ K \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \text{ különben.}$$

## HSI

A HSI (Hue, Saturation, Intensity) színmodell (ellentétben az RGB és a CMY színmodellekkel) jól illeszkedik a színek „leírásához”. A Hue a szín attribútuma, mely leír egy tiszta színt, a Saturation megadja annak telítettségét, az Intensity pedig a fényességét, világosságát. A HSI színmodellben tehát szétválik a szín- és az intenzitás-információ, így ideális a képfeldolgozó algoritmusokhoz. A HSI színteret általában két, alapjával egymásnak fordított kúppal reprezentálják, ahol a kúpok magasságvonala felel meg az intenzitás-skálának, egy magasságvonalra merőleges körlapokon a szögelfordulás a Hue komponensnek, az adott körlap középpontjától számított sugárirányú távolság pedig a Saturation-nak. Az RGB-ből a HSI-be történő átjárást az alábbi formulák adják meg:

$$H = \begin{cases} \theta, & \text{ha } B \leq G \\ 360 - \theta, & \text{különben,} \end{cases}$$

ahol

$$\theta = \arccos\left(\frac{(2R - G - B)/2}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right),$$

$$S = 1 - \frac{3 \cdot \min\{R, G, B\}}{R + G + B},$$



H komponens



S komponens



I komponens

2.3. ábra. A 2.2. ábrán bemutatott színes RGB kép HSI színmodellbe konvertáltjának komponensei monokróm képekként megjelenítve.

$$I = \frac{R+G+B}{3}.$$

A fordított irányú konverzió formulái:

$$R = I \cdot \left( 1 + \frac{S \cdot \cos H}{\cos(\pi/3 - H)} \right),$$

$$G = 3 \cdot I - R - B,$$

$$B = I \cdot (1 - S).$$

### YUV, YIQ, YCbCr

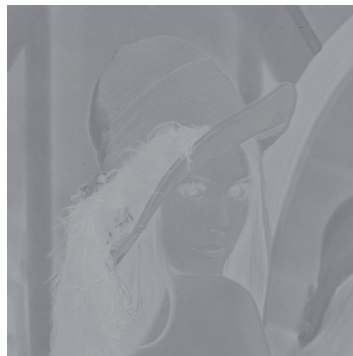
A fentiekén túl említést érdemelnek a színes televíziózásban és videózásban használt színmodellek, pl. a PAL/SECAM szabvány YUV színmodellje, az NTSC-nél használt YIQ és a széles körben (TIFF, JPEG, MPEG (DVD, digitális TV)) alkalmazott YCbCr színmodell. Ezekben közös az, hogy (a HSI-hez hasonlóan) elválik egymástól az intenzitás és a szín információ: mindhárom modellnél az „Y” komponens tartalmaz egy önmagában is „értékes” és „élvezhető” képet, amit a fekete-fehér televíziós készülékek jelenítenek meg, míg a fennmaradó két komponens hordozza a szín-információt. Az YUV, az YIQ és az YCbCr színmodellek és az RGB közötti konverziókat (eltérően a HSI-től) lineáris transzformációk írják le:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

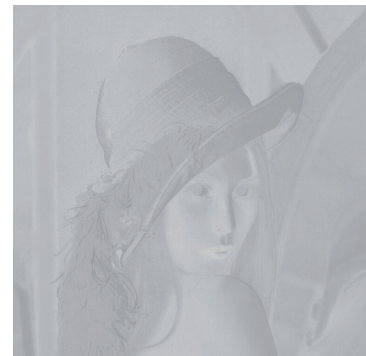
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.321 \\ 0.211 & -0.523 & 0.311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



Y komponens



Cb komponens



Cr komponens

2.4. ábra. A 2.2. ábrán látható színes RGB kép YCbCr színmodellbe konvertáltjának komponensei monokróm képekként megjelenítve.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.418 & -0.082 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

A fordított irányú konverziókat a fenti  $3 \times 3$  mátrixok inverzével történő szorzással hajthatjuk végre.

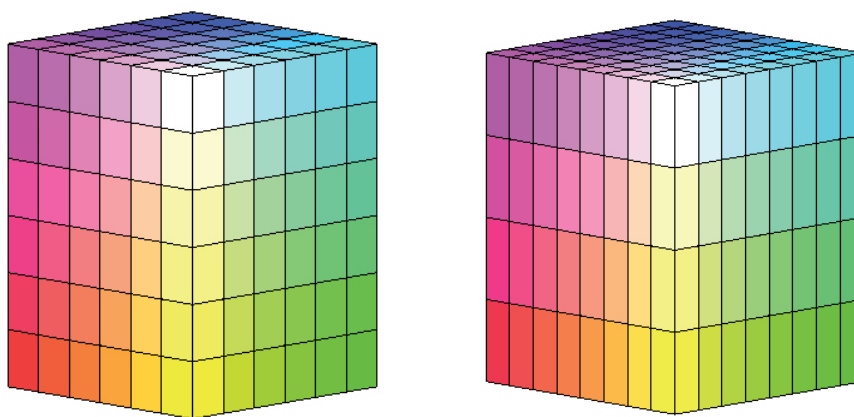
A 2.2.–2.4. ábrákon bemutatjuk egy színes kép komponenseit az RGB, a HSI és az YCbCr színmodellekben.

## 2.2. A színes képek feldolgozása

A színes képek feldolgozása alapvetően kétféleképpen történik:

- A *skaláris* vagy *komponensenkénti* feldolgozási sémánál a bemenő színes kép komponenseit mint egyedi monokróm képeket egymástól függetlenül dolgozzuk fel és az eredményül kapott képek lesznek az output kép komponensei.
- A *vektoros* feldolgozási módnál a bemeneti színes képek pontjait vektorokként kezeljük és transzformáljuk, nem bontjuk fel a képeket monokróm komponens-képekre.

Az alábbiakban áttekintjük a digitális képfeldolgozás néhány fontos területét színes képekre.



2.5. ábra. Az RGB színterének uniform kvantálásai a 256 színre: *safe RGB colors* (bal) és *3-3-2 RGB* (jobb).

### Kvantálás

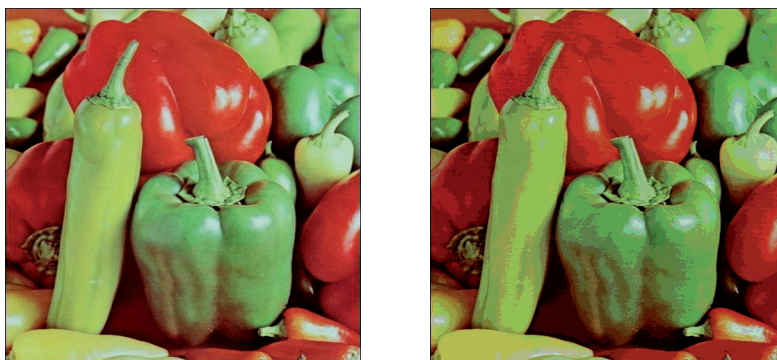
A színes képeken az egyes képpontokhoz tartozó 3– vagy 4–elemű vektorok nagyszámú lehetséges színárnyalatot írnak le (pl. a zínecsatornánként 256 értékkel bíró RGB full-color vagy



true-color képeken a lehetséges intenzitások száma  $2^{24} = 16\,777\,216$ ). Ezért számos területen (pl. a színes képek megjelenítésekor) szükség van az intenzitások, a kvantálási szintek számának csökkentésére úgy, hogy az ne járjon a képminőség érzékelhető romlásával.

A kvantálási szintek számának csökkentésére alapvetően két módszer kínálkozik: az *uniform*, amikor a színtér tartományát (pl. az RGB kockáját) az adott képtől függetlenül daraboljuk fel a kívánt számú egybevágó térrészre és az adott szegmensbe eső valamennyi színárnyalatot a térrészt reprezentáló színnel helyettesítjük. Amennyiben az RGB kockáját minden éle mentén 6-6 egyforma vastagságú szeletre vágjuk fel, úgy 216-féle szintet kapunk, melyek elnevezése: *safe RGB colors* vagy *safe Web colors* (ld. 2.5. ábra). Az RGB színtérnek egy másik uniform kvantálásával 8 bit/pixel-es, azaz 256 lehetséges szintet tartalmazó 3-3-2 RGB képekhez jutunk (ld. 2.5. ábra). Ekkor az RGB kockáját az R és a G tengelyek mentén 8-8 egyforma vastagságú szeletre vágjuk fel (így az R és a G színtengelyek 3-3 biten ábrázolhatók), míg a B tengely mentén mindössze 4 részre szeletelünk (így a négyféle kék 2 bittel írható le). A másik kettőhöz képest a kék alapszín háttérbe szorítását az indokolja, hogy az emberi szem kevésbé érzékeny a kékre, mint a vörös és a zöld alapszínekre: a kék színskálán a sötétebb intenzitások közül messze többet érzékelünk már feketének, mint a másik kettő esetében.

A 3-3-2 RGB uniform kvantálásra a 2.6. ábra mutat példát.



2.6. ábra. Példa 3-3-2 RGB uniform kvantálásra. A kiindulási RGB kép (bal) és annak kvantált változata (jobb).

A *nem-uniform* vagy *adaptív* kvantálás képfüggő, vagyis a színtér szeletelésekor az egyes szegmensek térfogatai különbözők lehetnek. A nem-uniform kvantálás menete:

1. Végezzünk statisztikai elemzést a kiindulási kép intenzitásaira.
2. A statisztika alapján határozzuk meg a redukált színtér elemeit.
3. Rendeljük hozzá az eredeti színtér valamennyi árnyalatához a redukált színtér elemeit, vagyis adjuk meg a CLUT-ot (color look-up table).
4. Állítsuk elő a kvantált képet (valamennyi pontjának régi intenzitásával címezzük meg a CLUT-ot, melynek az adott komponense megadja az illető pont új intenzitását).

Nem-uniform kvantálásra számos eljárást javasoltak. Itt most csak kettőt, a median cut és a K-means algoritmust vázoljuk.



2.7. ábra. Példa median cut algoritmusra.

A kiindulási RGB képre (bal) az eljárás eredménye 16 (középső) és 5 (jobb) klaszterre.

A klaszterező algoritmus egyben szegmentálásra is alkalmazható.

A *median cut* eljárás:

1. Határozzuk meg a színtérnek a képen előforduló színek befoglaló dobozát.
2. Rendezzük a dobozbeli színeket a leghosszabb doboz-oldal tengelye mentén.
3. Vágjuk szét az aktuális dobozt a mediánnál, vagyis úgy, hogy mindkét részbe azonos számú képpont színe essen.
4. Ismételjük az eljárást a 2. lépéstől, amíg el nem értük az előre adott doboz-számot (pl. 256-ot).
5. Rendeljük hozzá minden dobozhoz az átlagszínét.

A median cut eljárásra a 2.7. ábra mutat példát.

A *K-means* algoritmus:

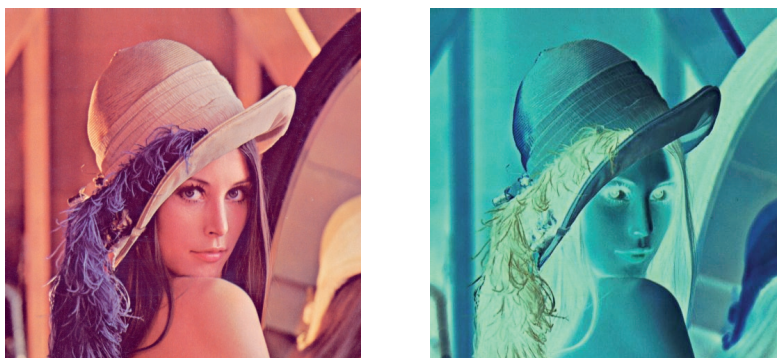
1. Adjunk meg előre adott  $K$  számú elemet kezdeti centroidokként (mint a szegmensek, klaszterek reprezentáns elemeit).



2.8. ábra. Példa K-means algoritmusra.

A kiindulási RGB képre (bal) az eljárás eredménye 8 (középső) és 4 (jobb) klaszterre.

Az eljárás szegmentálásra is alkalmazható.



2.9. ábra. Példa negációra. A színkomponensenként 256-árnyalatot tartalmazó RGB full-color kép (bal) és annak színes negatívja (jobb). Az input kép  $(r, g, b)$  intenzitású pontjához rendelt új szín a  $(255 - r, 255 - g, 255 - b)$   $(r, g, b = 0, 1, \dots, 255)$ .

2. Minden elemet rendeljünk a legközelebbi centroid klaszteréhez.
3. Számoljuk újra a klaszterek centroidjait.
4. Ismételjük a 2. és a 3. lépést mindaddig, míg elmozdulnak a centroidok.

A K-means algoritmus hatását a 2.8. ábra mutatja be.

Mielőtt a kvantálás területétől elköszönnénk, megemlítenéd, hogy a kvantálási szintek számának elégtelenségéből fakadó sávosodás színes képek esetében is csökkenthető mester-séges zaj hozzáadásával, a *dithering* technikával.

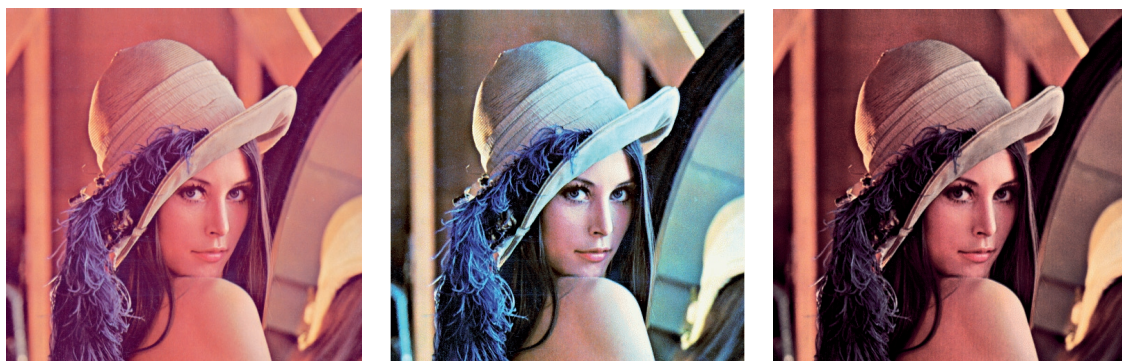
### Pont-operációk

A pont-operációknál az eredménykép valamely pontjának intenzitása csak az input kép ugyanazon pozíciójának intenzitásától függ, vagyis egy pont-operációt egy intenzitás-átmenet függvény ír le, ahol a vízszintes tengelyen az input kép intenzitásait, a függőlegesen pedig az output kép kvantálási szintjeit tüntetjük fel. A digitális (diszkrét) képek pont-operációihoz tartozó függvények egy-egy keresőtáblával (look-up-table, LUT) adhatók meg, mely táblát a bemenő



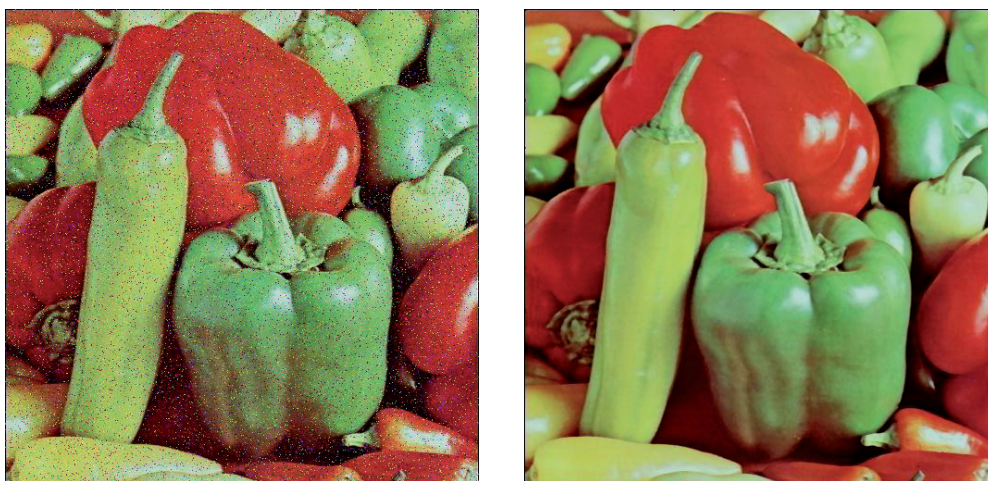
2.10. ábra. A 256 szürkeségi szinttel kvantált Lena kép (bal) és az álszínezett RGB kép (jobb). Az input kép  $g$  szürkeségi árnyalatához rendelt szín a  $(T_R(g), T_G(g), T_B(g))$   $(g = 0, 1, \dots, 255)$ , ahol  $T_R, T_G$  és  $T_B$  az egyes színkomponensekhez tartozó pont-operátorok.





2.11. ábra. A színes RGB Lena-kép (bal) hisztogram-kiegyenlítése mindhárom komponensen (középső) és a HSI-be konvertált kép I-csatornájának hisztogram kiegyenlítettje (jobb). Megállapíthatjuk, hogy a középső képen sérült a „színhűség”, míg csupán az intenzitás csatorna transzformációja nem deformálta a színeket.

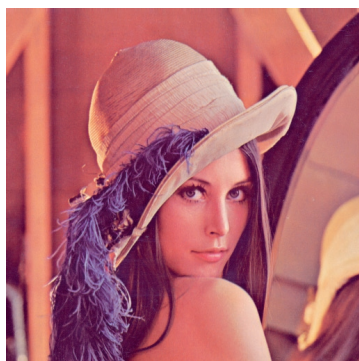
kép intenzitásaival címezzük meg és a megcímzett elem értéke a megfelelő kimenő intenzitás. A 2.9. és a 2.10. ábra két egyszerű pont-operációra, a negációra és az álszínezésre mutat példát.



2.12. ábra. Példa színes képek szűrésére. A só és bors típusú zajjal terhelt kép (bal) és azon a medián-szűrés eredménye (jobb).

A képfüggő pont-operációk meghatározása számos esetben a hisztogramon, vagyis a képen előforduló intenzitások relatív gyakoriságán alapul. A nevezetes hisztogram-alapú transzformációk (pl. hisztogram-széthúzás, hisztogram-kiegyenlítés, hisztogram-specifikáció) végrehajthatók színtonként vagy csak valamely releváns csatornán. A 2.11. ábra a hisztogram-kiegyenlítésre mutat példát abban az esetben, amikor az RGB kép valamennyi komponensére (egymástól függetlenül) hajtunk végre műveletet és azt is, amikor a HSI színmodellbe konvertált képnek csak a harmadik, az intenzitás csatornáját transzformáljuk (és a másik két komponenst változatlanul hagyjuk).

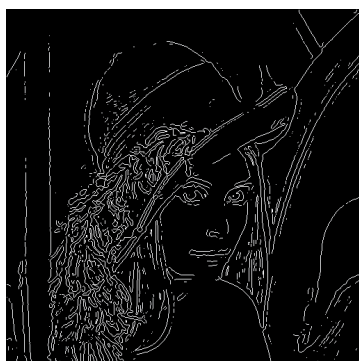




kiindulási kép



egyesített éltérkép



élek az R komponensen



élek a G komponensen



élek a B komponensen

2.13. ábra. Példa éldetektálásra. Az egyesített éltérkép a három színkomponensen (a Marr–Hildreth algoritmussal megtalált) éltérképek összegképe.

### Szűrés, simítás, él-detektálás

A pont-operációkhoz hasonlóan, színes képekre is alkalmazhatók a monokróm (szürkeárnyaltos) képekre megismert módszerek, viszont figyelembe kell venni azt, hogy nem alkalmas valamennyi színmodell a feldolgozásra. Pl. az RGB színmodellben valamennyi színkomponens egymástól független transzformációja után az eredménykomponensekből összeállított kép általában nem teljesíti az elvárásainkat (ld. 2.11. ábra). Érdekesebb áttérni olyan színmodellekre, ahol a szín- és az intenzitás-információ szeparált.

Színes képeken az éldetektálás akkor lesz korrekt, ha két külön éltérképet állítunk elő (a szürkeárnyaltos képekre ismert technikákkal, pl. a Canny éldetektátorral), melyek az intenzitás-éleket és a szín-éleket tartalmazzák. A két éltérkép egyesítése történhet pontonkénti (súlyozott) átlagolással, összegzéssel (1. vektornorma), maximumképzéssel ( $\infty$ . vektornorma), vagy akár a négyzetösszegek négyzetgyökének meghatározásával (2. vektornorma).

Színes képek szűrésére és éldetektálására a 2.12. és a 2.13. ábrákkal adunk példákat.

### Szegmentálás

Színes képek szegmentálásakor ugyancsak ki lehet használni azt a tényt, hogy az intenzitás-információ mellett szín-információval is rendelkezünk. Amennyiben a képen található objektumok színtartományai diszjunktak, úgy klaszterező eljárások (mint pl. a kvantáláskor emlí-

tett K-means algoritmus) is alkalmazhatók a szürkeárnyaltos képeknél megismert módszerek mellett. Itt most csak egyetlen példát mutatunk be, melyen a szín-szeletelés technikáját alkalmaztuk (ld. 2.14. ábra).



2.14. ábra. Szegmentálás szín-szeleteléssel. A kiindulási RGB képen (bal) az RGB egységkocka  $(0.7, 0.15, 0.2)$  pontja köré rajzolt  $0.2$ -sugarú gömbbe eső színeket az eredményképen (jobb) pirossal, a többi színárnyalatot pedig szürkével jelenítettük meg.

### Kódolás, tömörítés

Színes képek tömörítése (a tárigény csökkentése) történhet a kvantálási szintek számának csökkentésével (pl. RGB full-color reprezentáció helyett 256-színű palettás/indexelt reprezentációra való áttéréssel).

A színes álló- vagy mozgóképek kódolásakor és veszteséges tömörítésekor kihasználható az a tény is, hogy az emberi szem a színek változására kevésbé érzékeny, mint a fényességére. Így pl. a mozgóképeknél használt YUV, YIQ vagy YCbCr színmodellekben a szín-információt hordozó utolsó két komponenst ritkábban mintavételezik, mint az intenzitást/fényességet hordozó Y csatornát.

Színes képekre a JPEG tömörítő eljárás először a fenti (színt és intenzitást szeparáló) színmodellek valamelyikébe konvertál, majd az intenzitásnál a szín-információt tartalmazó komponenseket nagyobb tömörítési arányt (és egyben nagyobb veszteséget eredményező) kvantáló táblákat alkalmaz.

A színes képek feldolgozása iránt érdeklődők figyelmébe Gonzalez & Woods, valamint Sharma kézikönyveit ajánljuk [24, 61].

## 3. fejezet

# Matematikai morfológia

A *morfológia* a biológiának az élőlények alakjával foglalkozó ága, míg a Matheron és Serra által megalapozott *matematikai morfológia* (*mathematical morphology*) a kepanalízishez biztosít hasznos eszközöket [42, 56].

Bináris képeken a matematikai morfológia műveleteit halmazműveletekkel írjuk le, ahol halmazként elemi minták is szerepelnek.

A jelen fejezetben ismertetjük a matematikai morfológia alpműveleteit, a dilatációt és az eróziót, majd az azokból építkező morfológiai operátorok közül a nyitást, a zárást, a morfológiai szűrést, a hit-or-miss transzformációt, a morfológiai vékonyítást és a morfológiai vázat. Helyszűke miatt nem vállalkozhatunk további morfológiai operátorok és algoritmusok bemutatására, így ki kell hagynunk pl. a határkivonást, a régió feltöltést, az összefüggő komponensek meghatározását, a morfológiai vastagítást, a konvex burok morfológiai közelítését és a morfológiai váztisztítást. Az érdeklődő olvasó Gonzalez és Woods kézikönyvében [24] megtalálja a felsoroltakat.

A morfológiai műveleteket kiterjesztették többszintű, szürkeárnyaltos képekre is. A jelen fejezetben kitérünk a szürkeárnyaltos dilatációra, erózióra, nyitásra, zárásra, morfológiai szűrésre, morfológiai gradiensre, csúcs- és völgydetektorra.

A morfológiai műveletek jelölése a szakirodalomban nem egységes. A jelen fejezetben a Gonzalez és Woods kézikönyvének [24] jelölésrendszerét követtük.

### 3.1. Alpműveletek: dilatáció és erózió

A matematikai morfológia műveletei a  $d$ -dimenziós ( $d \geq 1$ ) euklidészi tér tetszőleges részhalmozain értelmezettek. Legyenek  $X$  és  $Y$  a képpontok  $V$  halmazának részhalmozai. (2D kép esetén  $V = \mathbb{Z}^2$ , 3D-ben pedig  $V = \mathbb{Z}^3$ ).

Az  $X$  ponthalmaz *komplementerét* jelölje  $X^c$ , ahol:

$$\begin{aligned} X^c &= \{ p \mid p \notin X \} \\ &= V \setminus X. \end{aligned}$$

Az  $X$  ponthalmaz  $a$ -val való *eltolása* a következőképpen definiált:

$$(X)_a = \{ x+a \mid x \in X \},$$

ahol  $a \in V$  és „+” a komponensenkénti/koordinátánkénti összeadást jelöli.

Az  $Y$  pontthalmaznak (az origóra való) *tükrözését* az alábbi formula adja meg:

$$\hat{Y} = \{ -y \mid y \in Y \},$$

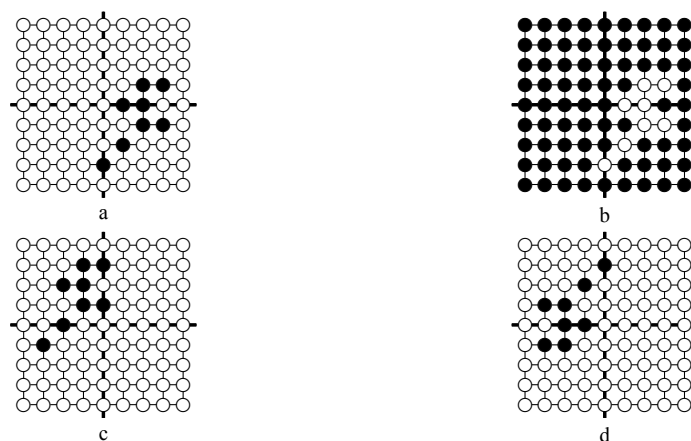
ahol a „-” a komponensenkénti „-1”-gyel való szorzást jelöli. Az  $Y$  pontthalmaz *szimmetrikus*, ha  $Y = \hat{Y}$ .

A fenti három segédműveletre a 3.1. ábra ad példákat.

Az  $X$  pontthalmaznak az  $Y$ -nal való *dilatációját* (*dilation*) a következő formula írja le:

$$\begin{aligned} X \oplus Y &= \{ a \mid (\hat{Y})_a \cap X \neq \emptyset \} \\ &= \{ a \mid [(\hat{Y})_a \cap X] \subseteq X \}, \end{aligned}$$

ahol az  $Y$  pontthalmazt *szerkesztőelemnek* (*structuring element*) nevezik. A szerkesztőelem hasonló szerepet játszik, mint a konvolúciós maszk: a szerkesztőelem koordinátarendszerének origóját eltoljuk  $V$  minden elemére és az adott elemet felvesszük a dilatált halmazba, ha a szerkesztőelemnek legalább egy pontja  $X$ -beli ponttal kerül fedésbe. Könnyen belátható, hogy ha  $Y$  tartalmazza az origót, akkor tetszőleges  $Y$ -ra:  $X \subseteq X \oplus Y$ , vagyis a dilatáció „hízal”.

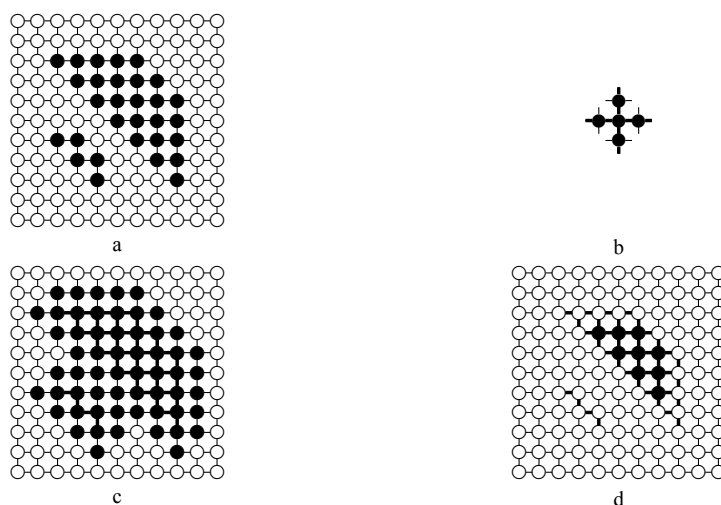


3.1. ábra. Példák a morfológiai segédműveletekre. Az  $X \subseteq \mathbb{Z}^2$  halmaz (a), ahol  $X$  elemeit „•”, a  $\mathbb{Z}^2 \setminus X$ -be esőket pedig „○” szimbólumok jelölik. A vastag vonalak az origóban (a (0,0) pontban) metszik egymást és az ábrázolt  $9 \times 9$ -es részleten kívül nincs  $X$ -beli pont. A három segédművelet eredménye:  $X^c$  (b),  $(X)_a$  a  $a = (-3, 2)$  esetén (c) és  $\hat{X}$  (d).

A dilatáció duálisát, az *eróziót* (*erosion*), az alábbi formula adja meg:

$$\begin{aligned} X \ominus Y &= \{ a \mid (Y)_a \subseteq X \} \\ &= (X^c \oplus \hat{Y})^c, \end{aligned}$$

vagyis a  $V$  halmaz egy pontja akkor kerül be az erodált halmazba, ha a szerkesztőelem koordinátarendszerének origóját az adott pontba eltolva minden egyes  $Y$ -beli pont  $X$ -belivel kerül fedésbe. Könnyen belátható, hogy ha  $Y$  tartalmazza az origót, akkor tetszőleges  $Y$ -ra:  $X \ominus Y \subseteq X$ , vagyis az erózió „fogyaszt”.



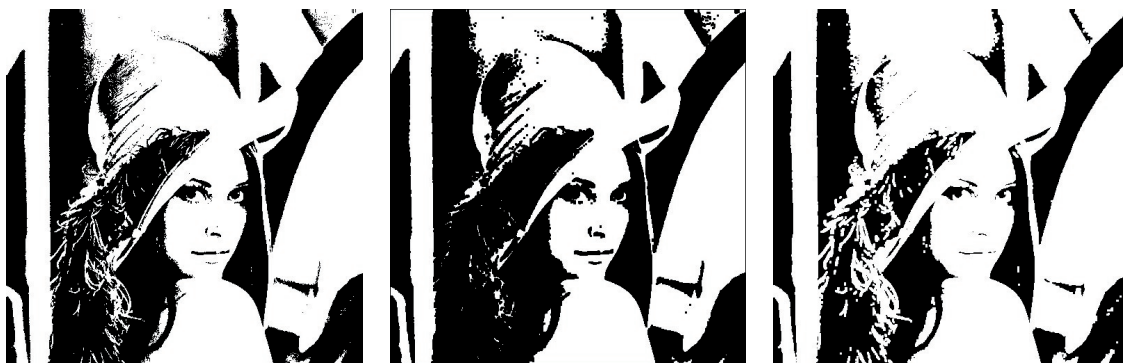
3.2. ábra. Példák dilatacióra és erózióra. Az  $X \subseteq \mathbb{Z}^2$  halmaz vagy bináris kép (a), melynek elemeit illetve fekete pontjait „•” szimbólumok jelölik. Az  $Y$  (szimmetrikus) szerkesztőelem (b) elemeit „•”-ek jelölik és a vastag vonalak az origóban (a  $(0,0)$  koordinátájú pontban) metszik egymást. Az  $X \oplus Y$  dilatació (c)  $X$  hízlalását, az  $X \ominus Y$  erózió (d) pedig  $X$  fogyását eredményezi. A 4-szomszédságnak megfelelő szerkesztőelem mellett a dilatació feketévé változtatja mindazon pontokat, amelyeknek van fekete 4-szomszédja, az erózió pedig csak azokat a fekete pontokat hagyja meg, amelyeknek az összes 4-szomszédja fekete volt. (A (c) és a (d) képeken azok a vonaldarabok vastagok, amelyek 4-szomszédos  $X$ -beli pontokat kötnek össze.)

A dilatació és az erózió hatását a 3.2. és a 3.3. ábrával illusztráljuk.

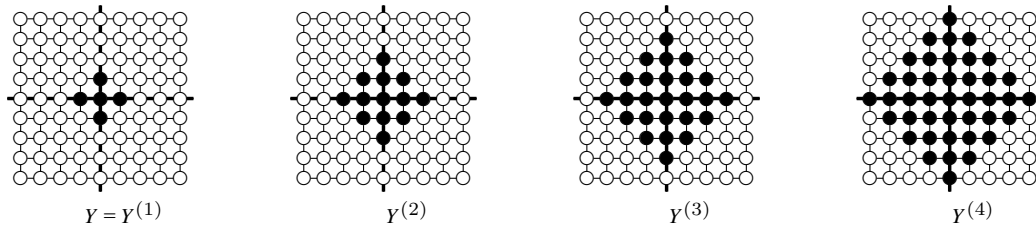
Bevezetjük az *iterált szerkesztőelem* fogalmát és az *iterált dilatació* valamint az *iterált erózió* műveleteket.

Legyen  $Y \subset V$  egy tetszőleges szerkesztőelem ( $V = \mathbb{Z}^2$  vagy  $\mathbb{Z}^3$ ) és jelölje  $O_V$  az origót (ami  $(0,0)$ , ha  $V = \mathbb{Z}^2$ , és  $(0,0,0)$ , ha  $V = \mathbb{Z}^3$ ). Az  $Y$  elem  $k$ -adik iteráltját jelölje  $Y^{(k)}$ , ahol:

$$Y^{(k)} = \begin{cases} \{O_V\}, & \text{ha } k = 0 \\ Y^{(k-1)} \oplus Y, & \text{ha } k \geq 1 \end{cases}$$



3.3. ábra. Egy bináris kép (bal) dilataciója (középső) és eróziója (jobb), ahol a fekete pontok tartoznak a ponthalmazhoz és az alkalmazott szerkesztőelem egy  $3 \times 3$ -as négyzet.



3.4. ábra. Példa iterált szerkesztőelemre. Az  $Y \subseteq \mathbb{Z}^2$  elembe és annak iteráltjaiba eső pontokat „•” szimbólumok jelölik. A vastag vonalak az origóban metszik egymást.

Iterált szerkesztőelemre a 3.4. ábrával adunk példát.

Az  $X$  ponthalmaznak az  $Y$  szerkesztőelemmel való iterált dilatacióját a következő formula írja le:

$$X \oplus_k Y = \begin{cases} X, & \text{ha } k = 0 \\ (X \oplus_{k-1} Y) \oplus Y, & \text{ha } k > 0 \end{cases}.$$

Az iterált erózió hasonlóképpen definiálható:

$$X \ominus_k Y = \begin{cases} X, & \text{ha } k = 0 \\ (X \ominus_{k-1} Y) \ominus Y, & \text{ha } k > 0 \end{cases}.$$

Könnyen belátható, hogy az erózióra és a dilatacióra teljesülnek alábbi tulajdonságok tetszőleges  $X$ ,  $Y$  és  $Z$  halmazok esetén:

$$\begin{aligned} (X \oplus Y) \oplus Z &= X \oplus (Y \oplus Z), \\ (X \ominus Y) \ominus Z &= X \ominus (Y \oplus Z). \end{aligned}$$

A dilatació és az erózió meghatározásából az alábbi két azonosság is következik:

$$\begin{aligned} X \oplus \{O_V\} &= X, \\ X \ominus \{O_V\} &= X. \end{aligned}$$

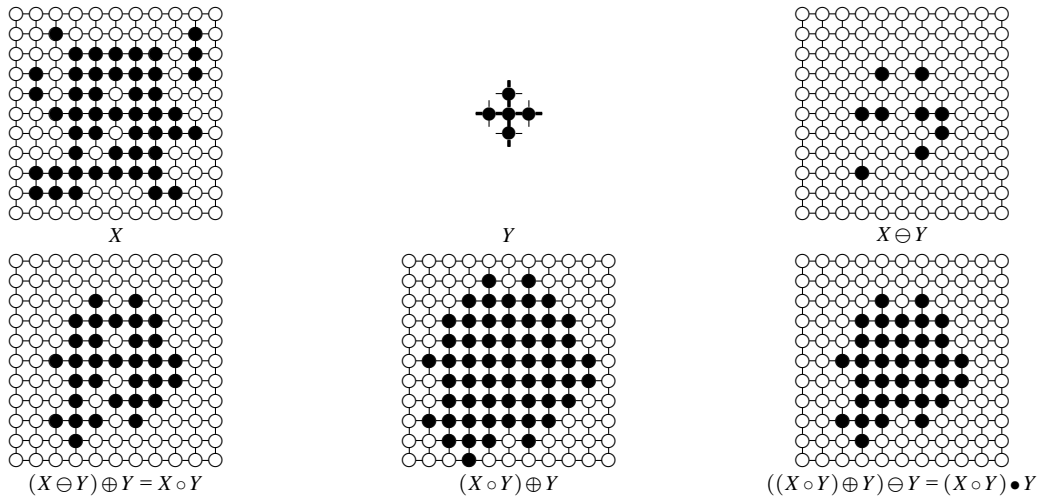
A fentiek alapján (pl. teljes indukcióval) könnyen beláthatók a következők:

$$\begin{aligned} X \oplus_k Y &= X \oplus Y^{(k)}, \\ X \ominus_k Y &= X \ominus Y^{(k)}. \end{aligned}$$

Megállapíthatjuk, hogy a „nagy méretű” szerkesztőelemekkel való dilatació és erózió kiváltható „kisméretűekkel” végrehajtott iterált műveletekkel.

## 3.2. Nyitás, zárás, morfológiai szűrés

A dilatacióból és erózióból, mint elemi építőkövekből számos további morfológiai operátor konstruálható.



3.5. ábra. Példa nyitásra, zárásra és morfológiai szűrésre, ahol az  $Y$  szerkesztőelem a 4-szomszédságnak felel meg. Az aktuális halmaz elemeit (vagy az aktuális kép fekete pontjait) „•” szimbólumok jelölik.

Az  $X$  ponthalmaznak az  $Y$  szerkesztőelemmel való nyitását (*open*) a következő formula definiálja:

$$\begin{aligned} X \circ Y &= \bigcup \{ (Y)_a \mid (Y)_a \subset X \} \\ &= (X \ominus Y) \oplus Y, \end{aligned}$$

azaz a nyitás egy erózió és egy dilatació egymás utáni végrehajtását jelenti. A nyitást a fenti formula első sora alapján úgy is elképzelhetjük, hogy az  $X$  halmaz megad a kifestőkönyvben egy szegmenst, az  $Y$  szerkesztőelem pedig leírja az ecset méretét és alakját, a nyitás eredménye pedig a szegmens „szabályos” kifestésének felel meg (ahol a szegmenst maximálisan kifestettük anélkül, hogy kifutottunk volna belőle).

A nyitás duálisa a zárás (*closing*), ahol a dilataciót követi az erózió:

$$\begin{aligned} X \bullet Y &= (X \oplus Y) \ominus Y \\ &= (X^c \circ \hat{Y})^c. \end{aligned}$$

A nyitás és a zárás a következő tulajdonságokkal bír:

1.  $X \circ Y \subseteq X$  és  $X \subseteq X \bullet Y$ .
2. Ha  $X_1 \subset X_2$ , akkor  $X_1 \circ Y \subseteq X_2 \circ Y$  és  $X_1 \bullet Y \subseteq X_2 \bullet Y$ .
3.  $(X \circ Y) \circ Y = X \circ Y$  és  $(X \bullet Y) \bullet Y = X \bullet Y$ .

Az 1. tulajdonság szerint nyitás „fogyaszt”, a zárás pedig „hízal”. A monotonitásról szól a 2. tulajdonság, míg a 3. azt mondja ki, hogy a nyitás és a zárás ismétlése hatástalan, azaz a műveletek idempotensek.





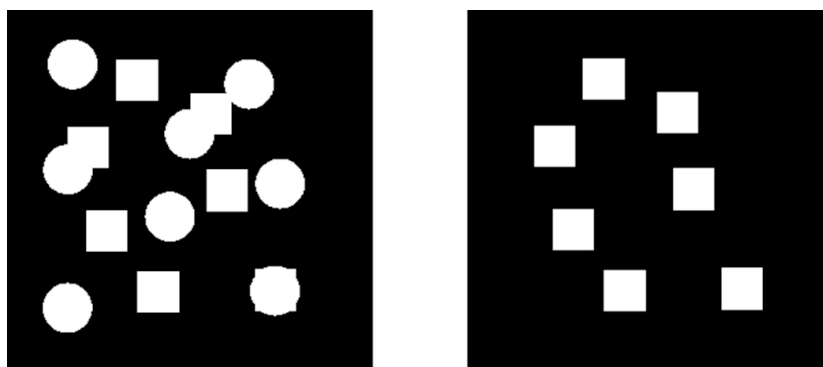
3.6. ábra. A nyitás (bal), a zárás (középső) és a morfológiai szűrés (jobb) eredménye a 3.3. ábrán látható bináris képre, ahol a fekete pontok tartoznak a ponthalmazhoz és az alkalmazott szerkesztőelem egy  $3 \times 3$ -as négyzet.

A fenti tulajdonságaik miatt a nyitás és a zárás szűrésre is használható. A morfológiai szűrést (*morphological filtering*) az

$$(X \circ Y) \bullet Y = (((X \ominus Y) \oplus Y) \oplus Y) \ominus Y$$

formula írja le, hatását a 3.5. ábrával szemléltetjük. Az ábra egyben a nyításra és a zárásra is példát ad.

A morfológiai szűrő egy erózióval indul, ami a bináris képekről törli a „kisméretű külső” zajokat. A kezdeti erózió mellékhatásaként a „kisméretű” (zajnak minősülő) üregek megnőnek és a megőrzendő objektumok „fogynak”. Az eróziót egy dilatáció követi, ami visszaállítja az objektumok és az üregek eredeti méretét. (Mivel az erózió teljesen törölte a „kisméretű külső” zajokat, így azok végképp eltűntek.) A szűrésnek a nyitás fázisa ezzel befejeződött. A zárást indító újabb dilatáció kitölti a „kisméretű” (zajnak minősülő) üregeket. A túlhízalt objektumok helyreállítása a zárást (és egyben a szűrést) befejező erózió feladata. A morfológiai szűrő hatását az alkalmazott  $Y$  szerkesztőelem megválasztása befolyásolja. A szerkesztőelem



3.7. ábra. Példa objektumok, foltok leválogatására nyitással. A kiindulási kép (bal) 33 pixel oldalhosszúságú négyzeteket és 20 pixel sugarú körlapokat tartalmaz. A nyitás egy 33 pixel oldalhosszúságú négyzettel mint szerkesztőelemmel alkalmas a körök eltávolítására (jobb). A nyitást kezdő erózió csak a négyzetek középpontjait hagyja meg, az azt követő dilatáció pedig visszahízlalja a középpontjaikból a négyzeteket.



mérete (és alakja) határozza meg azt, hogy mely objektumok és üregek minősülnek „kisméretűnek”.

A nyitásra, a zárásra és a morfológiai szűrésre a 3.6. ábra ad grafikus példát.

A 3.7. ábrán egy egyszerű példát láthatunk arra, hogy a nyitás (vagy a zárás) segítségével leválogathatunk bizonyos objektumokat a képről alakjuk és méretük alapján.

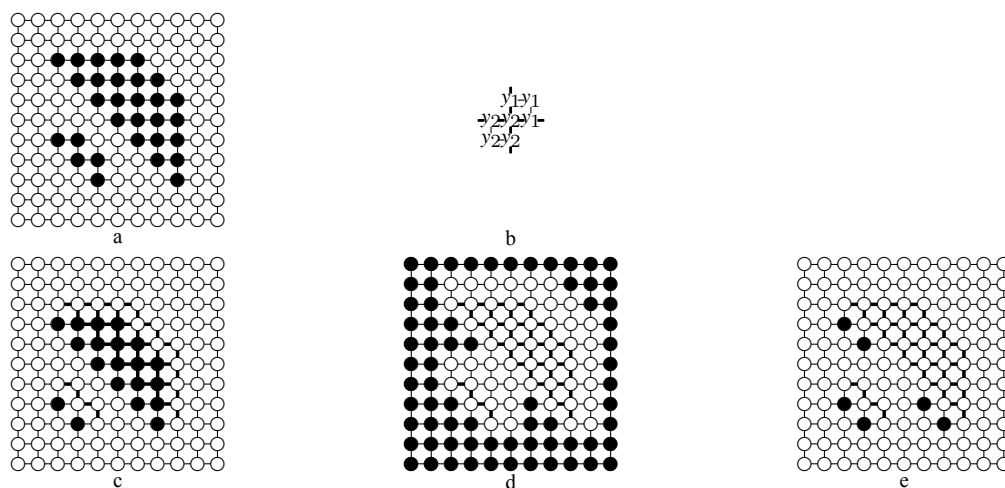
### 3.3. Hit-or-miss transzformáció

Bontsuk az  $Y$  szerkesztőelemet két diszjunkt részhalmazra:  $Y = Y_1 \cup Y_2$ , ahol  $Y_1 \cap Y_2 = \emptyset$ . Az  $X$  ponthalmaznak az  $Y$ -nal való *hit-or-miss transzformációját* (*hit-or-miss transform*) az alábbi formula adja meg:

$$\begin{aligned} X \circledast Y &= (X \ominus Y_1) \cap (X^c \ominus Y_2) \\ &= (X \ominus Y_1) \setminus (X \oplus \hat{Y}_2), \end{aligned}$$

vagyis a hit-or-miss transzformált halmaznak olyan pont lesz eleme, amellyel a szerkesztőelem origóját eltolva az  $Y_1$  által lefedett valamennyi pont  $X$ -be esik, ugyanakkor az  $Y_2$  által lefedett pontok között nincs  $X$ -beli.

A bit-minta illesztésére alkalmazható hit-or-miss transzformációra a 3.8. ábrával mutatunk példát.



3.8. ábra. Példa hit-or-miss transzformációra. Az  $X \subseteq \mathbb{Z}^2$  halmaz vagy bináris kép (a), melynek elemeit illetve fekete pontjait „•” szimbólumok jelölik. Az  $Y = Y_1 \cup Y_2$  szerkesztőelem (b), ahol az  $Y_1$  halmaz elemeit „ $y_1$ ”-ek, az  $Y_2$ -belieket pedig „ $y_2$ ”-k jelölik és a vastag vonalak az origóban metszik egymást. A (c) képen látható  $X \ominus Y_1$  eredménye, míg (d) mutatja az  $X^c \ominus Y_2$  képet. A végeredmény,  $X \circledast Y = (X \ominus Y_1) \cap (X^c \ominus Y_2)$  (e)-n látható. (A (c)–(e) képeken azok a vonaldarabok vastagok, amelyek 4-szomszédos  $X$ -beli pontokat kötnek össze.)

### 3.4. Morfológiai vékonyítás

A vékonyítás mint a vázkijelölés egyik módszere (ld. a 4. fejezet) a matematikai morfológia eszközeivel is leírható. Az iteratív objektum redukció egy iterációs lépés során (az objektumok határáról) eltávolítandó pontokat a hit-or-miss transzformáció jelöli ki. Az  $X$  halmaznak az  $Y$  szerkesztőelemmel való vékonyítása formálisan:

$$\begin{aligned} X \otimes Y &= X \setminus (X \circledast Y) \\ &= X \cap (X \circledast Y)^c. \end{aligned}$$

Hangsúlyozandó, hogy a fenti formula csak a vékonyítás egy iterációs lépésére vonatkozik, amit addig kell ismételni, amíg  $X \circledast Y \neq \emptyset$ .

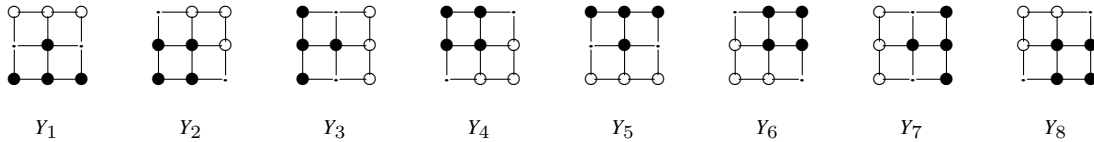
Amennyiben a vékonyítás az ún. irány-szekvenciális megközelítést követi (ld. a 4. fejezet), úgy egy iterációs lépés több fázisból (al-iterációból) áll, ahol az egymást követő fázisokhoz különböző szerkesztőelemek tartoznak.  $k$ -fázisú ( $k > 1$ ) vékonyítás esetén egy  $k$  tagból álló szerkesztőelem-rendszert kell megadni:

$$\mathcal{Y} = \{Y_1, Y_2, \dots, Y_k\},$$

ahol  $Y_i$  az  $i$ -edik fázis során alkalmazandó szerkesztőelem ( $1 \leq i \leq k$ ).

Az  $X$  halmaznak az  $\mathcal{Y}$  szerkesztőelem-rendszerrel történő vékonyításának egy iterációs lépését a következő formula írja le:

$$X \otimes \mathcal{Y} = ((\dots ((X \otimes Y_1) \otimes Y_2) \dots) \otimes Y_k).$$



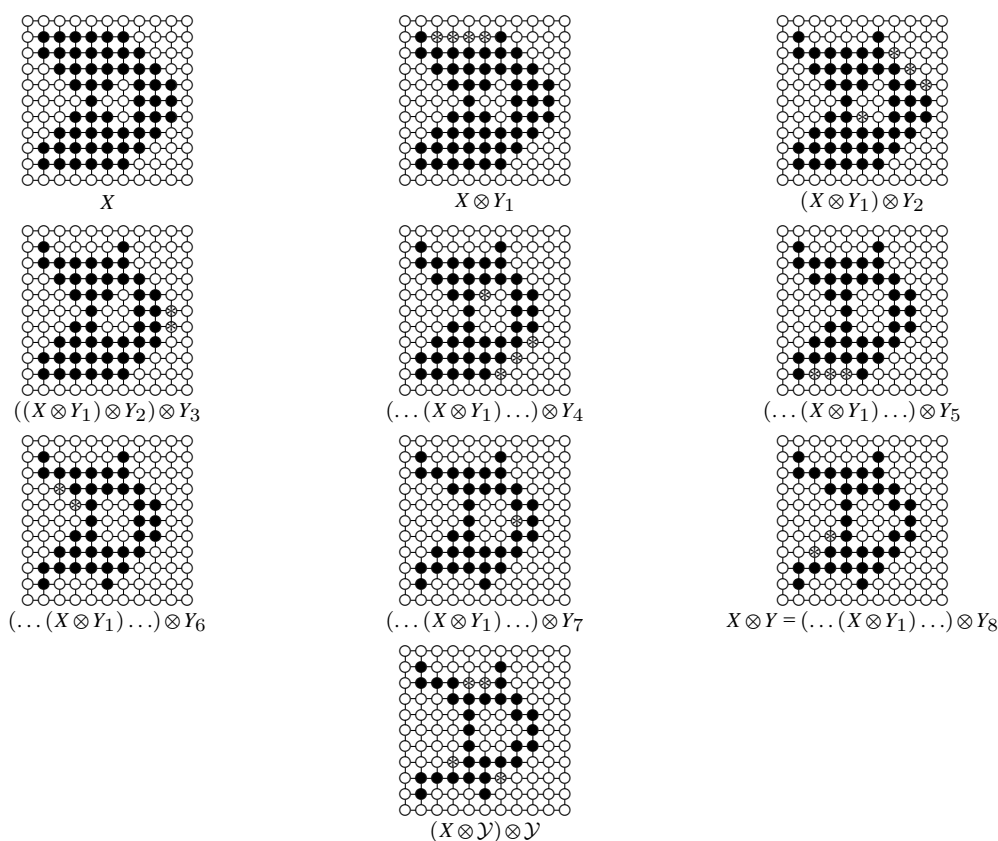
3.9. ábra. Az  $\mathcal{Y} = \{Y_1, Y_2, \dots, Y_8\}$  szerkesztőelem-rendszer 2D vékonyításra. Az  $Y_i = Y_{i1} \cup Y_{i2}$  szerkesztőelem a 8-fázisú vékonyítás  $i$ -edik fázisához tartozik ( $1 \leq i \leq 8$ ). Az  $Y_{i1}$ -beli elemeket „•”, az  $Y_{i2}$ -be esőket pedig „○” szimbólumok jelölik. Az origó minden esetben a  $3 \times 3$ -as környezet középpontja. Megjegyezzük, hogy az utolsó hat szerkesztőelem az első kettőnek elforgatott vátozata, ahol a forgatási szögek:  $90^\circ$ ,  $180^\circ$  és  $270^\circ$ .

A morfológiai vékonyítást (pontosabban annak egy lépését) képműveletként is megfogalmazhatjuk: a  $\mathcal{P} = (V, m, n, B)$  egy bináris képen az  $\mathcal{Y}$  szerkesztőelem-rendszerrel adott vékonyítás egy lépése a  $\mathcal{P}' = (V, m, n, B \otimes \mathcal{Y})$  képet eredményezi.

A morfológiai vékonyítás kívánt hatása a szerkesztőelem-rendszer alkalmas megválasztásával érhető el. A 3.9. ábrán látható 2D szerkesztőelem-rendszerrel végrehajtott 8-fázisú vékonyításra a 3.10. ábra mutat példát.

### 3.5. Morfológiai váz

A váz (mint egy gyakran alkalmazott régió-alapú alakleíró jellemző, ld. 4. fejezet) kifejezhető a matematikai morfológia műveleteivel is.



3.10. ábra. Példa morfológiai vékonyításra. Az eljárás a 3.9. ábrán bemutatott  $\mathcal{Y} = \{Y_1, \dots, Y_8\}$  szerkesztőelem-rendszerrel dolgozik. Az adott halmazba tartozó  $\mathbb{Z}^2$ -beli pontoknak „•”, míg a halmazon kívülieknek „◦” szimbólumok felelnek meg. Az adott fázisban törölt pontokat még a „\*” szimbólummal is megjelöltük.

Diszkrét terek (pl.  $\mathbb{Z}^2$  és  $\mathbb{Z}^3$  esetében) a morfológiai váz meghatározásához olyan  $Y$  (szimmetrikus) szerkesztőelemet szoktak feltételezni, ami az origó középpontú, 1-sugarú hipergömb egy közelítése.

A morfológiai vázat a következő formula adja meg:

$$S(X) = \bigcup_{k=0}^{\infty} S_k(X) = \bigcup_{k=0}^{\infty} \{(X \ominus_k Y) \setminus [(X \ominus_k Y) \circ Y]\}.$$

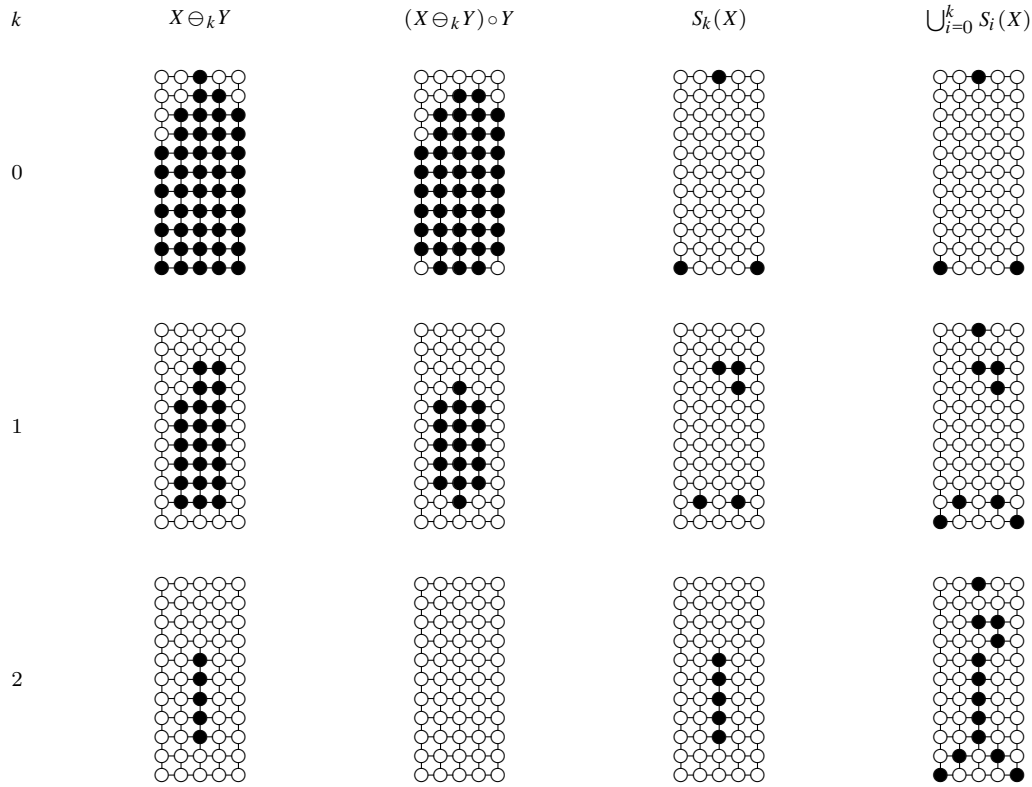
Véges  $X$  halmazra az  $S(X)$  váz véges számú halmaz egyesítésével áll elő, vagyis:

$$S(X) = \bigcup_{k=0}^K S_k(X) = \bigcup_{k=0}^K \{(X \ominus_k Y) \setminus [(X \ominus_k Y) \circ Y]\},$$

ahol

$$K = \max\{k \mid (X \ominus_k Y) \neq \emptyset\}.$$

Morfológiai vázra a 3.11. ábrával mutatunk egy 2D-s példát. Megállapíthatjuk, hogy a példában szereplő objektum váza nem lesz összefüggő. Ez egyrészt abból ered, hogy diszkrét



3.11. ábra. Példa morfológiai vázra, ahol az  $Y$  szerkesztőelem a 4-szomszédságnak felel meg. A bal felső képen látható a kiindulási  $X = X \ominus_0 Y$  halmaz, melynek  $S(X) = \bigcup_{i=0}^2 S_i(X)$  váza a jobb alsó képen található, ahol  $S_i(X) = (X \ominus_i Y) \setminus [(X \ominus_i Y) \circ Y]$ .

pontrácson értelmezett a kép, másrészt pedig a szerkesztőelem és annak iteráltjai nem közelítik jól a körlapokat.

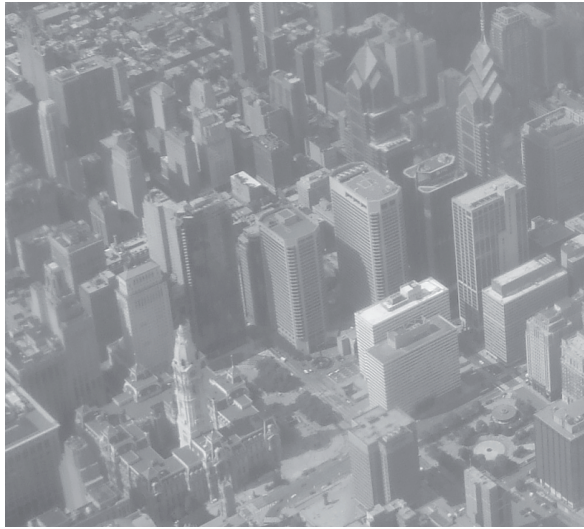
A morfológiai váz egy érdekes tulajdonsága, hogy (a szerkesztőelem megválasztásától függetlenül) a kiindulási  $X$  halmaz rekonstruálható a morfológiai váz  $S_k(X)$  részhalmazainak ismeretében:

$$X = \bigcup_{k=0}^K S_k(X) \oplus_k Y.$$

### 3.6. Morfológia többszintű képekre

A ponthalmazok (vagy bináris képek) dilataciója és eróziója, valamint a belőlük építkező összetett műveletek kiterjeszthetők többszintű, szürkeárnyaltos képekre is. A kiterjesztésre két megközelítést is javasoltak:

- Ha a szerkesztőelem továbbra is csak egy ponthalmaz, tartomány, akkor *lapostetejű* (flat topped) szerkesztőelemről és operátorokról beszélünk.
- Ha a szerkesztőelemhez tartozó pontoknak értéke, intenzitása is van (többszintű miniképeknek tekinthetők), akkor a szerkesztőelem és a művelet *nem-lapostetejű* (non-flat top-



A

 $A \oplus S$  $A \ominus S$ 

3.12. ábra. Egy többszintű kép dilatációja és eroziója, ahol a lapostetejű szerkesztőelem egy  $5 \times 5$ -ös négyzet.

ped). Megjegyezzük, hogy a lapostetejű szerkesztőelem egy olyan nem-lapostetejűnek tekinthető, ahol a szerkesztőelem valamennyi pontjának 0 az értéke.

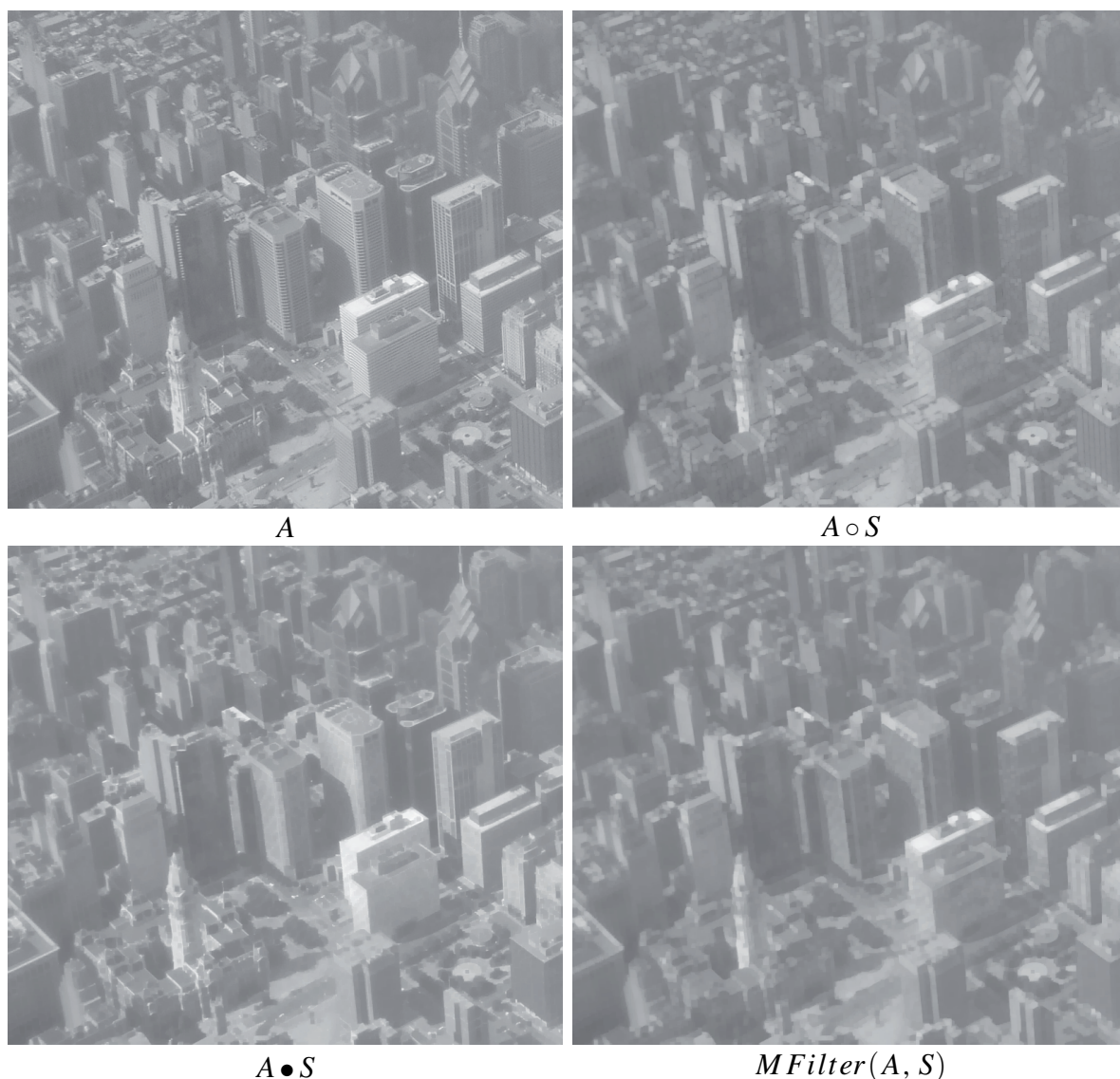
Legyen  $A = [a(i, j)]$  egy többszintű kép és jelölje a lapostetejű  $S$  szerkesztőelem tartományát (az  $S$ -be eső pontok halmazát)  $D_S$ . Ekkor a  $B = A \oplus S = [b(i, j)]$  többszintű képet eredményező dilatációt a következő formula írja le:

$$b(i, j) = \max_{(u,v) \in D_S} \{a(i-u, j-v)\}.$$

A  $C = A \ominus S = [c(i, j)]$  többszintű erózió formulája:

$$c(i, j) = \min_{(u,v) \in D_S} \{a(i+u, j+v)\}.$$





3.13. ábra. Nyitás, zárás és morfológiai szűrés eredménye egy többszintű képre, ahol a lapostetejű szerkesztőelem egy  $5 \times 5$ -ös négyzet.

Könnyen belátható, hogy a fenti formulák érvényesek bináris képekre is: Egy képpont akkor lesz 1-értékű a dilatált képen, ha az illető pontra a tükrözött szerkesztőelem origóját illesztve szerkesztőelem pont 1-értékű képponttal kerül fedésbe. Hasonlóképpen, egy képpont értéke akkor lesz 1 az eróziót követően, ha az illető pontra az origóval illesztett szerkesztőelem valamennyi pontja 1-értékű képponttal kerül fedésbe.

Nem-lapostetejű  $S = [s(u, v)]$  szerkesztőelemre a  $B = A \oplus S = [b(i, j)]$  dilatáció és a  $C = A \ominus S = [c(i, j)]$  erózió formulái:

$$b(i, j) = \max_{(u,v) \in D_S} \{a(i-u, j-v) + s(u, v)\},$$

$$c(i, j) = \min_{(u,v) \in D_S} \{a(i+u, j+v) - s(u, v)\}.$$





A

 $Grad(A, S)$  $Laplace(A, S)$ 

3.14. ábra. A morfológiai gradiens és a morfológiai Laplace-transzformáció egy többszintű képre, ahol a lapostetejű szerkesztőelem egy  $5 \times 5$ -ös négyzet.

A két alapl művelet között dualitás, vagyis az

$$A \ominus S = (A^c \oplus \hat{S})^c$$

egyenlőség a többszintű képek esetén is teljesül, viszont ekkor az  $A^c = [a^c(i, j)]$  komplementer-képre és az  $\hat{S} = [\hat{s}(u, v)]$  tükrözött szerkesztőelemre az alábbi formulák érvényesek:

$$a^c(i, j) = -a(i, j)$$

és

$$\hat{s}(u, v) = s(-u, -v).$$

Többszintű képek dilataciójára és eróziójára a 3.12. ábra mutat példát.

Többszintű képek esetén a nyitást, a zárást és a morfológiai szűrést rendre az alábbi formulák írják le:

$$A \circ S = (A \ominus S) \oplus S,$$

$$A \bullet S = (A \oplus S) \ominus S,$$

$$MFilter(A, S) = (A \circ S) \bullet S.$$

Többszintű képek nyitására, zárására és morfológiai szűrésére a 3.13. ábra mutat példát.

A többszintű képeken értelmezett nyitás és zárás tulajdonságainak bemutatása előtt szükségünk van a képek közötti „ $\prec$ ” precedencia reláció bevezetésére. Az  $A_1 = [a_1(i, j)]$   $m_1 \times n_1$ -es és az  $A_2 = [a_2(i, j)]$   $m_2 \times n_2$ -es képek re  $A_1 \prec A_2$ , ha  $m_1 \leq m_2$ ,  $n_1 \leq n_2$  és  $a_1(i, j) \leq a_2(i, j)$  ( $i = 1, \dots, m_1$ ,  $j = 1, \dots, n_1$ ).

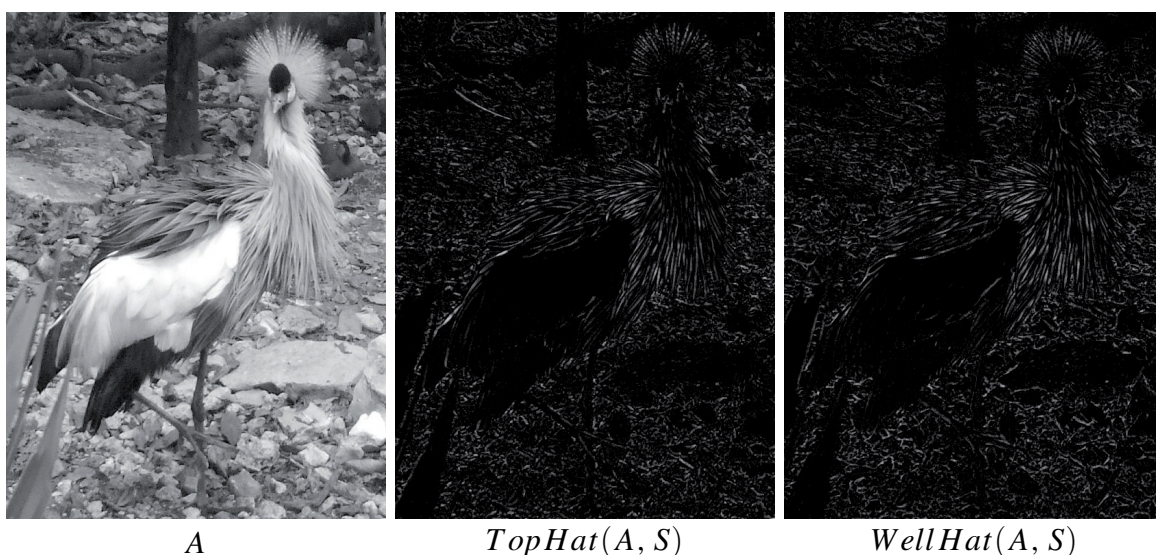
A műveletekre az alábbiak teljesülnek (tetszőleges  $A$  és  $B$  képekre, valamint  $S$  szerkesztőelemre):

1.  $A \bullet S = (A^c \circ \hat{S})^c$  és  $A \circ S = (A^c \bullet \hat{S})^c$  (vagyis a nyitás és a zárás egymás duálisai).
2.  $A \circ S \prec A$  és  $A \prec A \bullet S$  (azaz a nyitás „sötétít”, a zárás pedig „világosít”).
3. Ha  $A \prec B$ , akkor  $A \circ S \prec B \circ S$  és  $A \bullet S \prec B \bullet S$  (vagyis mindkét művelet monoton).
4.  $(A \circ S) \circ S = A \circ S$  és  $(A \bullet S) \bullet S = A \bullet S$  (vagyis mindkét művelet idempotens).

A többszintű képeken további négy (a binárisokra nem definiált) műveletet is javasoltak, melyek a *morfológiai gradiens* és a *morfológiai Laplace-transzformáció* (éldetektálásra), a *top-hat transzformáció* (csúcs-detektor, vagyis a környezetüket tekintve kiugróan „világos” pontok detektálására), valamint a *well-hat transzformáció* (völgy-detektor, vagyis a környezetüket nézve kiugróan „sötét” pontok detektálására):

$$\begin{aligned} \text{Grad}(A, S) &= (A \oplus S) - (A \ominus S), \\ \text{Laplace}(A, S) &= (A \oplus S) + (A \ominus S) - 2 \cdot A, \\ \text{TopHat}(A, S) &= A - (A \circ S), \\ \text{WellHat}(A, S) &= (A \bullet S) - A. \end{aligned}$$

A morfológiai gradiensre és a morfológiai Laplace-transzformációra a 3.14, a top-hat és a well-hat transzformációkra pedig a 3.15. ábra mutat példát.



3.15. ábra. A top-hat transzformáció és a well-hat transzformáció egy többszintű képre, ahol a lapostetejű szerkesztőelem egy  $5 \times 5$ -ös négyzet.

A többszintű képeken a matematikai morfológia eszköztárának két legfontosabb alkalmazási területe a *textúra-alapú szegmentálás* (*textural segmentation*) és a *granulometria* (*granulometry*). Alkalmasan megválasztott alakú és méretű szerkesztőelemekkel végrehajtott nyitással és zárással eltüntethetők, detektálhatók a képről adott alakú és mérethatár alatti világos ill. sötét foltok, továbbá meghatározható a foltok méret-eloszlása.

A matematikai morfológia és az alkalmazásai iránt érdeklődőknek Serra alapl műve [56] mellett számos kézikönyvet is ajánlhatunk [5, 22, 24, 25, 53, 63, 64].

## 4. fejezet

# Vázkijelölés

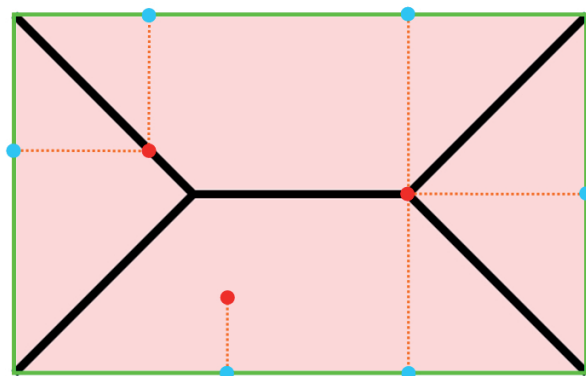
A váz egy gyakran alkalmazott régió–alapú jellemző, mely összegzi az objektumok alakját, formáját.

A jelen fejezetben a váz meghatározásainak és a folytonos váz tulajdonságainak bemutatása után ismertetjük a diszkrét (bináris) objektumokra a „vázszerű” jellemzők meghatározására adott három általános technikát: a távolság-információn alapulót, a mintavételezett határpontok mint generálopontok Voronoi-diagramjából származót és az objektum iteratív objektum redukcióján alapulót, a vékonyítást.

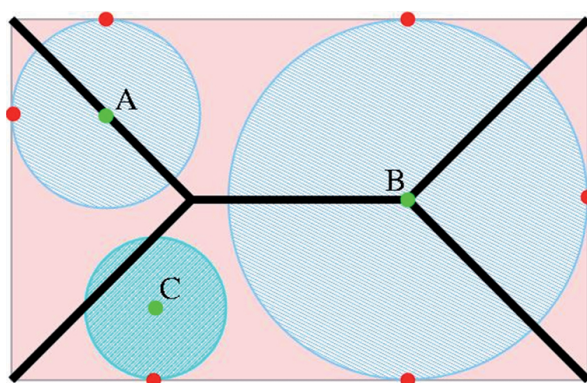
A tárgyalt módszerek számos olyan algoritmust (pl. távolságtérkép számítása, Voronoi-diagramok és a Delaunay felbontás meghatározása) is bemutatunk, amelyeket a képfeldolgozás számos más területén is gyakran alkalmaznak.

### 4.1. A váz meghatározása és tulajdonságai

A váz fogalma először Blum tanulmányaiban jelent meg [7, 8] mint a *középtengely–transzformáció* (*Medial Axis Transform, MAT*), vagy más elnevezéssel a *szimmetriatengely–transzformáció* (*Symmetry Axis Transform, SAT*) eredménye. A középtengely–transzformáció az objektum minden egyes pontjára megkeresi a hozzá legközelebbre



4.1. ábra. A váz a MAT szerint. A fekete pontokból álló váz pontjai legalább kettő legközelebbi határponttal bírnak.



4.2. ábra. A fekete pontokat tartalmazó vázat a maximális beírható körlapok középpontjai alkotják.

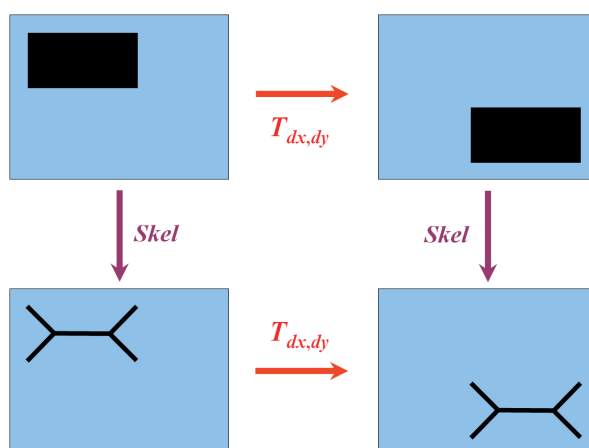
eső határponto(ka)t. Ha az eljárás egy belső pontra egynél több legközelebbi határpontot talál, akkor azt a vázhoz tartozónak, vázpontnak minősíti (ld. 4.1. ábra).

A vázat Blum egy szemléletes hasonlaltal, a *préritűz terjedésével* illusztrálta: Ha a vizsgált objektum határának minden pontját egyidejűleg meggyújtjuk és feltételezzük, hogy a tűzfrontok minden irányba egyenletes sebességgel terjednek, akkor a váz azokból a pontokból áll, ahol az objektum belsejében a tüzek találkoznak, kioltják egymást.

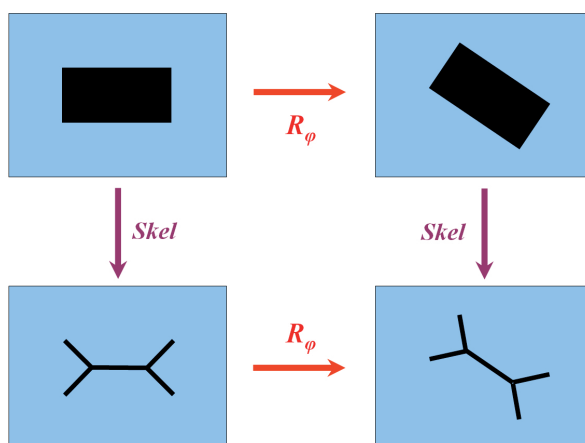
Ugyancsak Blum javasolta a váz *távolság-transzformációval* (*Distance Transform*) történő meghatározását. Az objektum minden pontjára kiszámítandó a tőle legközelebbre lévő határponttól vett távolsága, így megkapjuk a *távolságtérképet* (*distance map*). A vázat a távolságtérkép lokális maximumhelyei alkotják.

A vázra formális definíciót Calabi adott [16]: valamely 2-dimenziós objektum vázát azon pontok alkotják, melyek középpontjai az objektumba írt maximális nyílt körlapoknak (ld. 4.2. ábra). (Egy körlap nyílt, ha nem tartalmazza a határát, beírható az objektumba, ha a körlap részhalmaza az objektumnak és maximális beírható, ha nem valódi részhalmaza egyetlen beírható körlapnak sem.) Ez a meghatározás általánosítható magasabb dimenziókra is:  $dD$ -ben a nyílt körlapokat  $d$ -dimenziós nyílt hipergömbökkel kell helyettesíteni.

A váz a következő tulajdonságai miatt alkalmas *alakjellemzőnek* (*shape feature*):



4.3. ábra. A vázkijelölés invariáns az eltolásra.



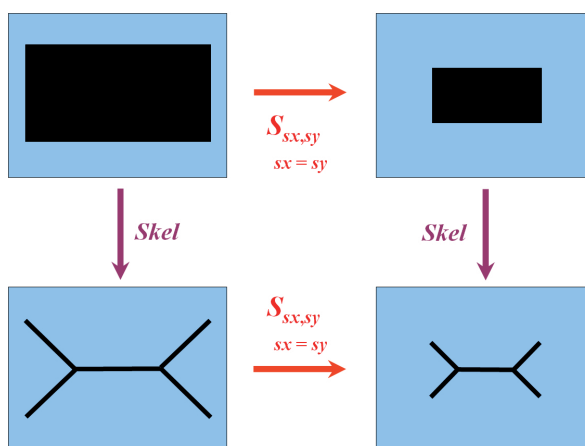
4.4. ábra. A vázkijelölés invariáns az elforgatásra.

- Az első fontos tulajdonság az, hogy a váz invariáns bizonyos geometriai transzformációkra. Könnyű belátni, hogy a váz csak olyan transzformációkra lehet invariáns, amelyeknél a hipergömbök képe hipergömb. Ilyenek pl. az *eltolás* (*translation*) (ld. 4.3. ábra), az *elforgatás* (*rotation*) (ld. 4.4. ábra), az *uniform skálázás* (*uniform scaling*) (vagyis a valamennyi tengely mentén ugyanazon faktorú skálázás) (ld. 4.5. ábra), vagy azok tetszőleges kompozíciója. A váz nem invariáns a nem-uniform skálázásra (ld. 4.6. ábra).

Invariancián azt értjük, hogy bármely  $T$  transzformációra és  $\mathcal{O}$  objektumra a transzformáció és a vázkijelölés tetszőleges sorrendben hajtható végre, vagyis

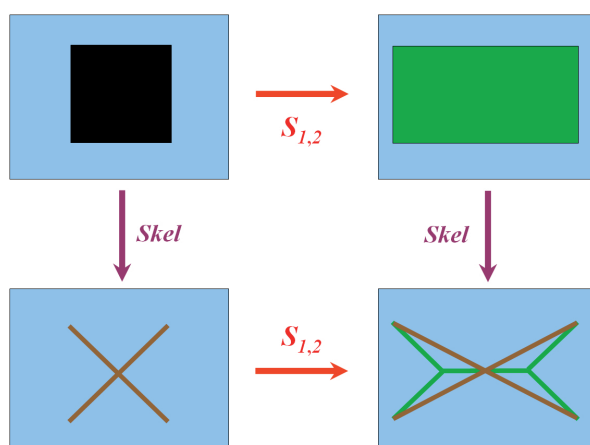
$$T(S(\mathcal{O})) = S(T(\mathcal{O})).$$

- A váz második lényeges tulajdonsága az, hogy reprezentálja az eredeti objektum topológiai struktúráját: Összefüggő objektum váza összefüggő, üreget vagy lyukat tartalmazó objektumnak a váza is üreges illetve lyukas (ld. 4.7. ábra) (Megjegyezzük, hogy



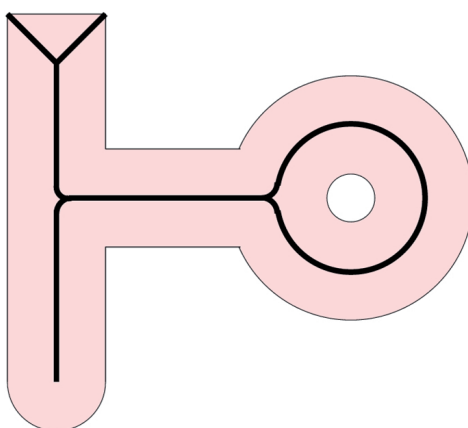
4.5. ábra. A vázkijelölés invariáns az uniform skálázásra.



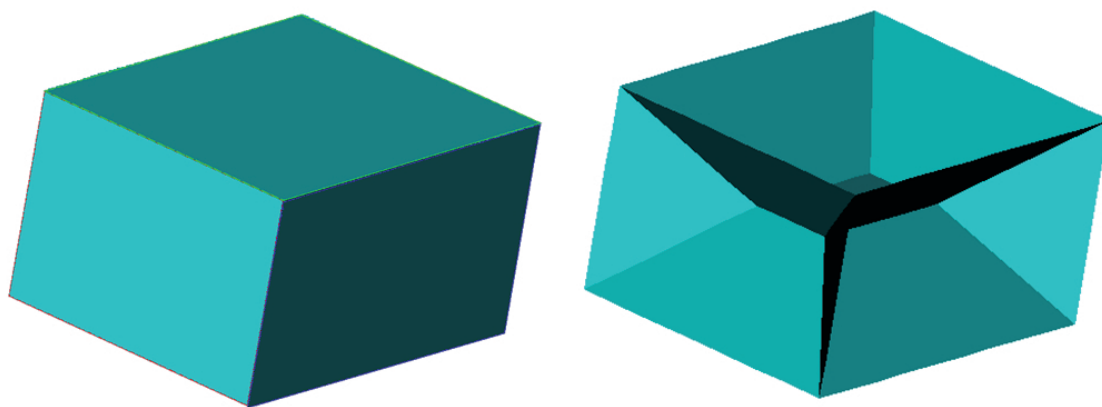


4.6. ábra. A vázkijelölés nem invariáns a nem-uniform skálázásra.

a folytonos 2-dimenziós váz csak akkor nem őrzi meg a topológiát, ha az objektum köríven érintkező szegmenseket tartalmaz.)

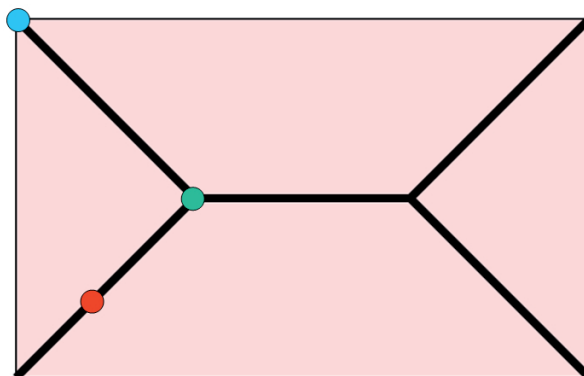


4.7. ábra. A váz reprezentálja a kiindulási objektum topológiai szerkezetét.



4.8. ábra. Egy 3-dimenziós tömör téglatest (bal) és váza (jobb), mely 2-dimenziós felszín-szegmenseket is tartalmazhat.





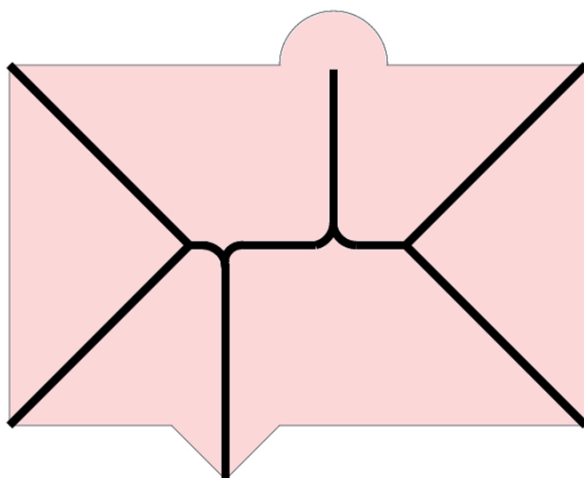
4.9. ábra. A 2-dimenziós váz háromféle pontja: vonal-végpont (kék), vonalpont (piros) és elágazási pont (zöld).

- A váz további jellemzésekor még két fontos tulajdonságra térünk ki. A váz egyrészt „vékony” (azaz egy pont vastagságú részekből tevődik össze), másrészt pedig egy erősen redukált szerkezet (jóval kevesebb pontból áll, mint a kiindulási objektum), ráadásul a váz elemeinek alacsonyabb a dimenziója is. A 2-dimenziós váz csak 1-dimenziós vonal-szegmensekből, görbékből áll, míg a 3-dimenziós objektumok váza 2-dimenziós felszín-szegmenseket is tartalmazhat (ld. 4.8. ábra).

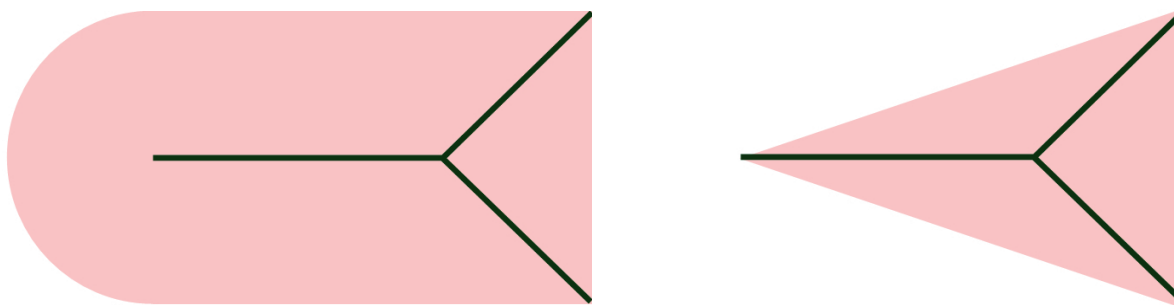
A 2-dimenziós objektumok váza mindössze háromféle pontot tartalmazhat: vonal-végpontokat, vonalpontokat és elágazási pontokat (ld. 4.9. ábra). Mivel a váz az eredeti kiterjedt objektumnál jóval kevesebb pontból áll, így a vázokkal reprezentált objektumokra drasztikusan csökken a kiértékelendő adatmennyiség.

A fenti tulajdonságok miatt a váz számos területen alkalmazható, pl. az *alakfelismerésben* (*pattern recognition*) [12] vagy a *képillesztés/képregisztráció* (*image matching/image registration*) (ld. 8. fejezet) problémájának megoldásakor is.

Nem hallgatható el a váz mint *régió-alapú alakjellemző* érzékenysége az objektum határának „kismértékű” megváltozására, még akkor sem, ha ez a stabilitási probléma a *határ-alapú*



4.10. ábra. Példa a váz zajérzékenységére.



4.11. ábra. Különböző objektumokhoz is tartozhat ugyanaz a váz.

jellemzőknél is jelentkezik. A 4.10. ábrán egy „kishibás” téglalap váza látható. Az objektumok „kismértékű” különbözősége a vázak „jelentős” eltérését vonja maga után. A váznak ez a hátrányos tulajdonsága az objektum határának *simításával* (*smoothing, filtering*) mint előfeldolgozással és/vagy a váz *tisztításával* (*pruning*) mint utófeldolgozó lépéssel orvosolható.

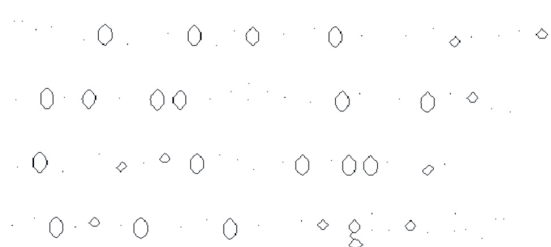
A folytonos váz jellemzését a reprodukálhatóság kérdésével zárjuk. Meghatározható-e egyértelműen az eredeti kiterjedt objektum a váz ismeretében?

A válasz nemleges (ld. 4.11. ábra), vagyis különböző objektumoknak megegyezhet a vázuk, így a vázkijelölés nem injektív – és ezáltal nem is bijektív – leképezés. A váz tehát csak akkor alkalmas az objektumok kódolására, ha minden egyes vázpontra (mint egy beírt maximális nyílt hipergömb középpontjára) ismert a hozzá tartozó hipergömb sugara is.

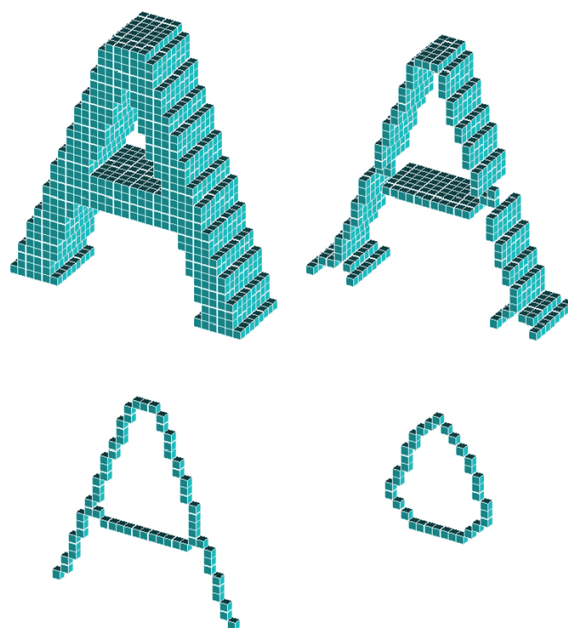
Diszkrét (digitális képeken található) objektumok esetén a pontos vázat csak egy diszkrét ponthalmazzal tudjuk közelíteni, illetve „vázszerű” alakleíró jellemzőket tudunk kivonni. Ilyen jellemzők 2-dimenziós objektumokra a *topológiai mag* (*topological kernel*) és a *kö-*

**"If you would know what the  
Lord God thinks of money,  
you have only to look at  
those to whom he gives it."**

"If you would know what the  
Lord God thinks of money,  
you have only to look at  
those to whom he gives it."



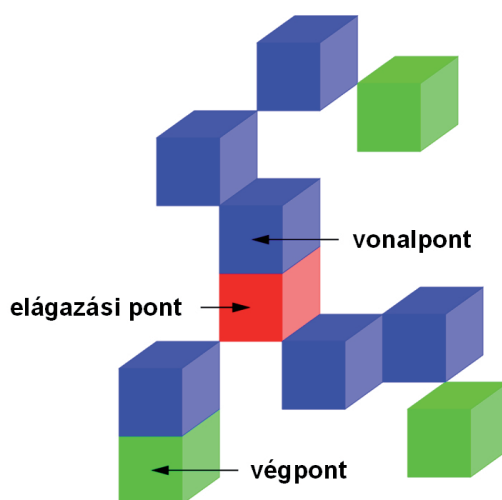
4.12. ábra. Egy 2-dimenziós bináris kép (fent), az objektumok középvonala (lent, bal) és topológiai magja (lent, jobb). Üreget nem tartalmazó objektumok esetében a topológiai mag egyetlen pont, az üreget tartalmazóké pedig egy zárt görbe (vagy zárt görbék egy pont vastag görbékkel összekötött szerkezete).



4.13. ábra. Egy 3-dimenziós objektumok (fent, bal) vázszerű jellemzői: középfelszín (fent, jobb), középvonal (lent, bal) és topológiai mag (lent, jobb).

zépvnal (centerline) (ld. 4.12. ábra), 3-dimenziós esetben pedig még a középfelszín (medial surface) (ld. 4.13. ábra).

A topológiai mag egy, a kiindulási objektummal topológiailag ekvivalens minimális pont-halmaz [35]. A topológiai mag nem őrzi meg az objektumok „alakját” (ld. 4.12. ábra), de mégis fontos szerepet játszik számos alkalmazásnál (pl. valamely szegmentáló algoritmus



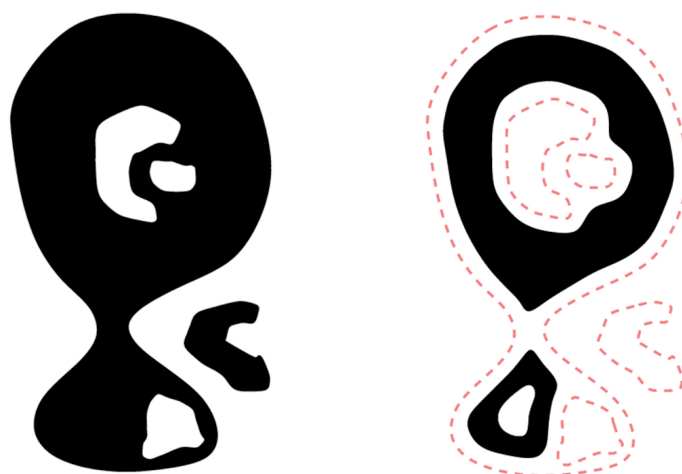
4.14. ábra. Példa 3-dimenziós középvonalra, melynek valamely eleme (egységkockája) végpont, ha pontosan egyetlen további kockával szomszédos (vagyis lapon, élen vagy csúcson osztozik), vonalpont, ha pontosan két szomszédja van és elágazási pont, ha kettőnél több szomszédal bír.

eredményének vizsgálatakor, objektum-címkézéskor vagy a képen található objektumok megszámlálásakor).

A 3-dimenziós objektumok középfelšíne (mint vázszerű jellemző) áll közel magához a vázhoz, mivel a 3-dimenziós váz 2-dimenziós felšín-szegmenseket is tartalmazhat. Cső-szerű objektumok esetén viszont a középvonal ad a gyakorlatban hasznosabb jellemzést. A 2-dimenziós esethez hasonlóan, a 3-dimenziós középvonalak is csak 3-féle pontot tartalmazhatnak: vonal végpontokat, vonalpontokat és elágazási pontokat (ld. 4.14. ábra).

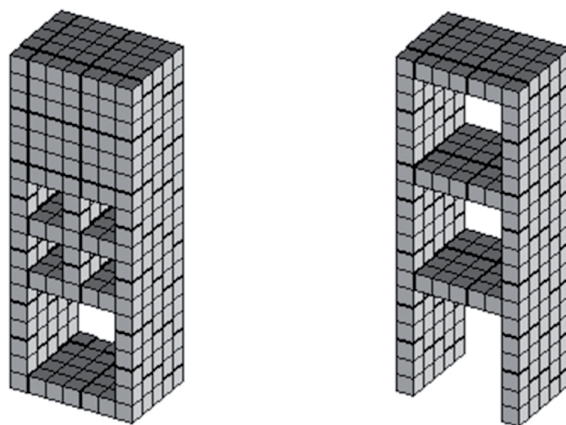
A vázkijelölés (*skeletonization*) feladata a diszkrét objektumok vázszerű jellemzőinek meghatározása. A vázkijelölő módszerekkel szemben az alábbi elvárásokat fogalmazták meg:

- A topológia elvárás: az eredményezett vázszerű jellemző legyen topológiaiilag ekvivalens (ld. később) a kiindulási objektummal.
- A geometriai kívánalom: a kapott szerkezet legyen egy pont vastag, helyezkedjen el az objektum közepén és legyen invariáns a legfontosabb geometriai transzformációkra (pl. eltolás, elforgatás, uniform skálázás).



4.15. ábra. Példa képredukcióra, mely nem őrzí meg a topológiát, mivel a kiindulási kép (bal) nagyobbik objektumát két részre szakította az eredményképen (jobb), a redukció a kisebbik objektumot teljesen törölte, összeolvasztott üregeket egymással és a háttérrel, továbbá egy új üreget is létrehozott.

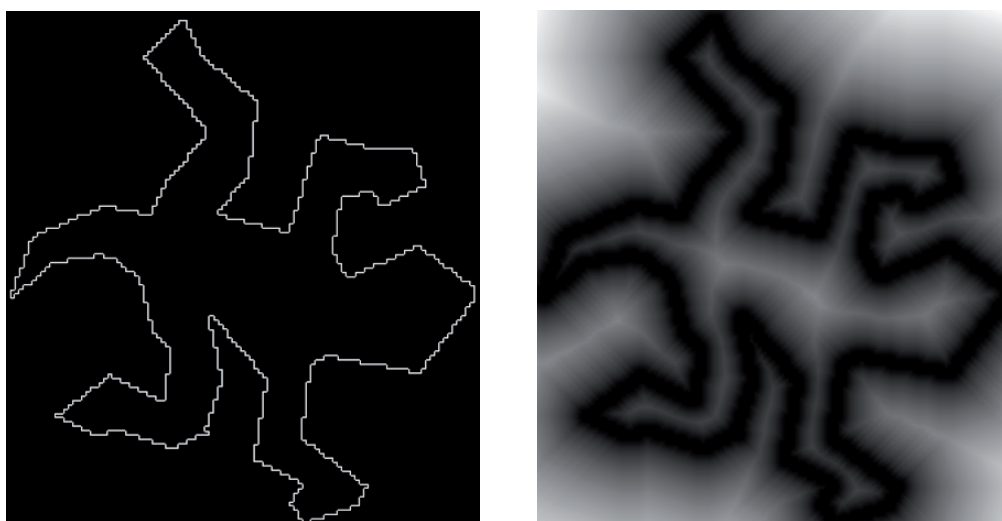
A digitális topológiával – sajnos helyhiány miatt – nem foglalkozhatunk behatóbban, így a topológiai kritériumot csak felvázolni tudjuk. (A digitális topológia íránt mélyebben érdeklődőknek olvasmányul Kong és Rosenfeld áttekintő cikkét [35] ajánljuk.) Könnyen belátható (bármelyik vázmeghatározás alapján), hogy egy objektum váza része a kiindulási objektumnak, így a vázkijelölés egy redukciós képművelet. Egy redukció akkor és csakis akkor őrzí meg a topológiát (akkor és csakis akkor marad az eredménykép topológiaiilag ekvivalens a kiindulásival), ha a bemenő kép egyetlen objektumát sem szakítja szét kettő vagy több részre, egyetlen objektumot sem töröl teljesen, nem olvaszt össze üregeket (az objektumokon mint kontinenseken belüli tavakat) sem egymással, sem pedig a háttérrel (maradva földrajzi hasonlatnál: az óceánok alkotta területtel) és új üreget sem hoz létre. A 2-dimenziós topológiamegőrző képredukciókra a 4.15. ábra mutat ellenpéldát.



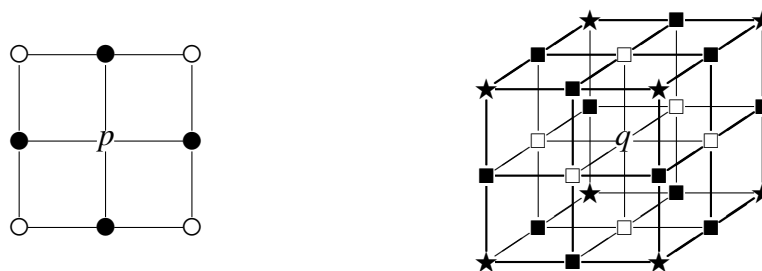
4.16. ábra. Példa 3-dimenziós képredukcióra, mely nem őrzi meg a topológiát. A kiindulási kép (bal) legalsó lyukja az eredményképen (jobb) összeolvadt a háttérrel, a redukció a négy kisebb lyukat összeolvasztotta egymással, felül pedig egy vadonatúj lyukat hozott létre.

A 3-dimenziós digitális képeken előkerül egy új fogalom, a lyuk (vagy alagút) is, melynek nincs megfelelője 2-dimenzióban. Formális definíció helyett most elégedjünk meg annyival, hogy a tórusz (vagy az amerikai fánk) egy lyukas objektum. A topológia-megőrző 3-dimenziós redukcióknál (a 2-dimenziós esetenél felsoroltakon túl) nem megengedett a lyukak összeolvasztása egymással vagy a háttérrel és egyetlen új lyuk létrehozása sem (ld. 4.16. ábra).

A következő három alfejezetben a vázkijelölés megoldásának három fő módszerével foglalkozunk.



4.17. ábra. Egy bináris kép, ahol a jellemző pontok a fehérek (bal) és annak távolságtérképe (jobb), ahol a nagyobb távolság-értékek világosabb intenzitással jelennek meg.



4.18. ábra. A  $\mathbb{Z}^2$  képrácson (vagy a vele duális négyzetmozaikon) a legfontosabb szomszédsági relációk a 4- és a 8-szomszédság. A  $p$  pont 4-szomszédainak  $N_4(p)$  halmaza tartalmazza  $p$ -t és a „●” pontokat, a 8-szomszédok  $N_8(p)$  halmaza pedig  $N_4(p)$  elemein túl tartalmazza a „○” pontokat is (bal). A  $\mathbb{Z}^3$  rács (vagy a vele duális kockamozaik leggyakrabban használt szomszédsági relációi a 6-, a 18- és a 26-szomszédság. A  $q$ -val 6-, a 18- és a 26-szomszédos pontok halmazát jelölje rendre  $N_6(q)$ ,  $N_{18}(q)$  és  $N_{26}(q)$ , ahol  $N_6(q)$  tartalmazza  $q$ -t és a „□” pontokat,  $N_{18}(q)$  tartalmazza  $N_6(q)$  mellett a „■” pontokat és  $N_{26}(q)$  tartalmazza  $N_{18}(q)$ -t és a „★” pontokat (jobb). Könnyen belátható, hogy a fenti relációk mindegyike reflexív és szimmetrikus.

## 4.2. Távolság-alapú vázkijelölés

A távolság-alapú vázkijelölés a *távolság-transzformáció* (distance transformation) alapul. A távolság-transzformáció bemenete egy  $A$  bináris kép, ahol az 1-értékű elemek a *jellemző pontok* (feature points), a kimenete pedig a  $B$  (nem-bináris) tömb, a *távolság-térkép* (distance map), ahol valamennyi pozíció értéke a megfelelő input képpont és a hozzá legközelebb eső jellemző pont távolsága a figyelembe vett távolság szerint, azaz

$$B(p) = \min\{d(p, q) \mid A(q) = 1\}.$$

Távolság-transzformációra a 4.17. ábra ad példát.

A távolság-transzformáció alapulhat az euklidészi távolságon, de akár az azt közelítő diszkrét távolságokon is. Megjegyezzük, hogy a diszkrét távolságok bizonyos esetekben nem metrikák. A diszkrét távolságokat származtathatjuk digitális képterek szomszédsági relációi-

4	3	2	1	2	3	4
3	2	1	0	1	2	3
2	1	0	1	0	1	2
2	1	0	1	1	0	1
1	0	1	2	2	1	0
1	0	1	2	3	2	1
0	1	2	3	4	3	2

2	2	1	1	1	2	2
2	1	1	0	1	1	2
2	1	0	1	0	1	1
1	1	0	1	1	0	1
1	0	1	1	1	1	0
1	0	1	2	2	1	1
0	1	1	2	2	2	2

4.19. ábra. Példa távolságtérképekre, ahol a 4-távolságot (bal) és a 8-távolságot (jobb) vettük figyelembe. A kiindulási kép mindkét esetben ugyanaz (a rózsaszín pozíciók a jellemző pontok), de a távolságtérképek különböznek egymástól.



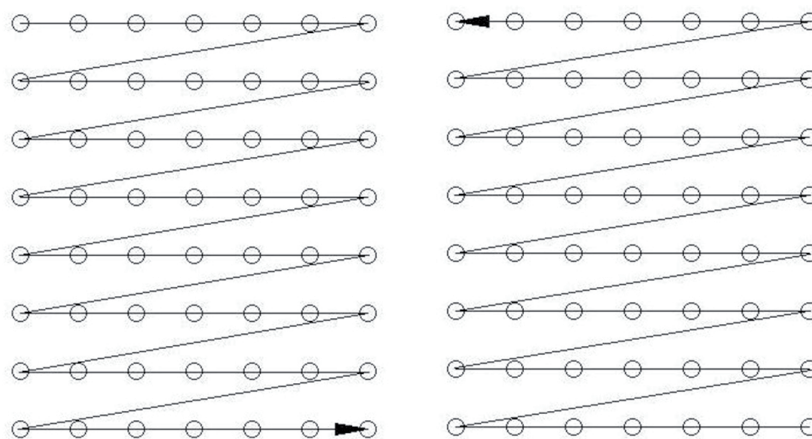
ból a szomszédok súlyozásával is. Formulák helyett a  $\mathbb{Z}^2$  és a  $\mathbb{Z}^3$  ortogonális képrácsokon a leggyakrabban alkalmazott *szomszédsági relációkat* a 4.18. ábra illusztrálja.

Az adott  $\mathbb{Z}^2$  vagy  $\mathbb{Z}^3$  rácson egy pontsorozat  $j$ -út ( $j = 4, 8, 6, 18, 26$ ), ha a sorozat szomszédos elemei  $j$ -szomszédosak. Egy  $j$ -út *hossza* az út elemszáma mínusz egy. Két pont  $j$ -*távolsága* az őket összekötő utak minimális hossza. Másképpen fogalmazva: a szomszédsági relációk elmozdulásokat adnak meg (ahol a szomszédokba való átlépés tekintett elmozdulásnak), a távolság pedig a minimális elmozdulások száma, amely a kérdéses egyik pontból a másikba visz. A 4- és 8-távolság szerint számított távolságtérképekre a 4.19. ábra ad példát.

Diszkrét tereken, képeken tehát a szomszédsági relációkból diszkrét távolságokat származtathatunk. A diszkrét távolságokra egy másik megközelítés a *chamfer távolság* [10], ahol az engedélyezett elmozdulásokhoz nemnegatív egész számokat, súlyokat rendelünk. Két pont közötti út költsége a benne szereplő elmozdulások összsúlya, a távolságuk pedig a köztük futó összes lehetséges út költségeinek minimuma.

A  $\mathbb{Z}^2$  rácson a 4-szomszédokba történő elmozdulás súlya  $d_1$ , a 8-szomszédokba (melyen nem 4-szomszédok) való elmozdulásoké pedig  $d_2$ . A  $\mathbb{Z}^3$  rácson a 6-szomszédokba történő elmozdulás súlya  $d_1$ , a 18-szomszédokba (melyen nem 6-szomszédok) az elmozdulások súlya  $d_2$ , a 26-szomszédokba (melyen nem 18-szomszédok) való elmozdulásokhoz pedig a  $d_3$  súlyokat rendeljük. A  $\mathbb{Z}^2$  rácson a  $(d_1, d_2) = (3, 4)$  választás jól közelíti az euklidészi távolság 3-szorosát (mivel  $\frac{d_2}{d_1} = \frac{4}{3} = 1,33 \dots \approx \sqrt{2} = 1,414 \dots$ ). A  $\mathbb{Z}^3$  rácson  $(d_1, d_2, d_3) = (3, 4, 5)$  a  $d_1=3, d_2=4$  és  $d_3=5$  választással szokás az euklidészi távolság 3-szorosát közelíteni (mivel  $\frac{d_2}{d_1} = \frac{4}{3} = 1,33 \dots \approx \sqrt{2} = 1,414 \dots$  és  $\frac{d_3}{d_2} = \frac{5}{4} = 1,25 \dots \approx \sqrt{3} = 1,732 \dots$ ).

Borgefors megmutatta, hogy a chamfer távolságokon alapuló távolság-transzformáció lineáris,  $\mathcal{O}(n)$  időben meghatározható, ahol  $n$  a kép pontjainak a száma [10]. Az alábbiakban Borgefors módszerét mutatjuk be 2- és 3-dimenziós képekre.



4.20. ábra. A pásztázás előre (bal) és a pásztázás vissza (jobb) által követett bejárások.

**Algoritmus:**  $(d1, d2)$ -chamfer távolságtérkép számítására

*Input:* az  $A = [a(i, j)]$   $n1 \times n2$ -es bináris kép

*Output:* a  $B = [b(i, j)]$   $n1 \times n2$ -es távolságtérkép

// inicializálás

for  $j = 1$  to  $n2$  do

  for  $i = 1$  to  $n1$  do

    if  $a(i, j) = 1$  then

$b(i, j) = 0$

    else

$b(i, j) = \infty$

// pásztázás előre

for  $j = 1$  to  $n2$  do

  for  $i = 1$  to  $n1$  do

$b(i, j) = \min\{$

$b(i-1, j-1)+d2, b(i-1, j)+d1, b(i-1, j+1)+d2,$

$b(i1, j-1)+d1, b(i, j) \}$

// pásztázás hátra

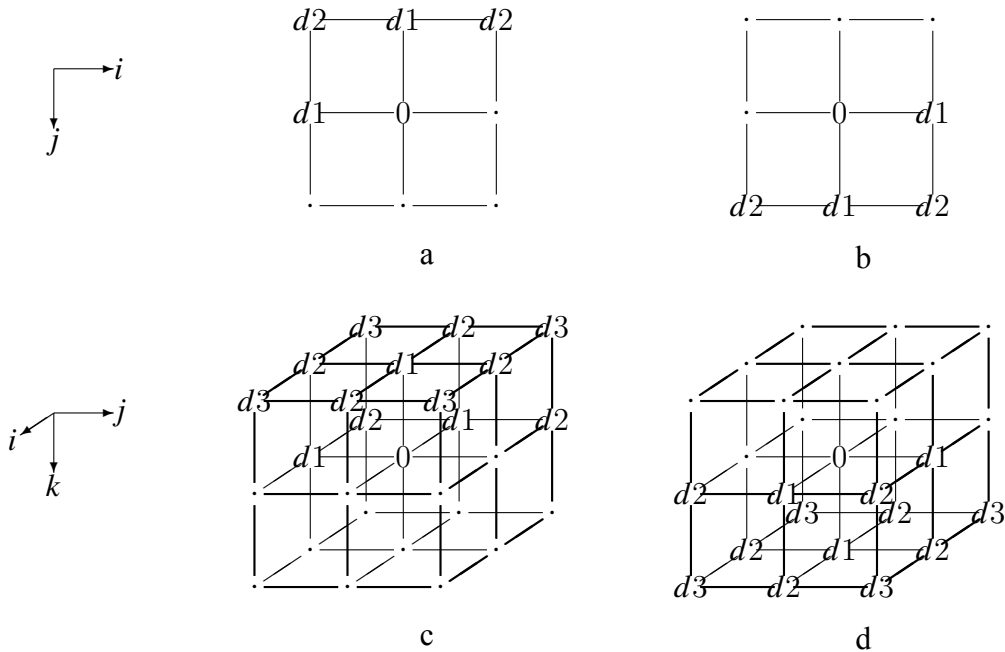
for  $j = n2$  downto 1 do

  for  $i = n1$  downto 1 do

$b(i, j) = \min\{$

$b(i, j), b(i, j+1)+d1,$

$b(i+1, j-1)+d2, b(i+1, j)+d1, b(i+1, j+1)+d2 \}$



4.21. ábra. Lokális környezetek és súlyok chamfer távolságtérképekhez. A figyelembe vett szomszédság 2-dimenzióban a pásztázás előre fázisban (a) és a pásztázás vissza menet során (b). 3-dimenzióban a pásztázás előre fázis súlyai (c), a pásztázás vissza menetben pedig (d). A fenti szomszédsági maszkokat középpontjukkal az éppen meglátogatott elemre kell helyezni és a súlyok által lefedett elemek súlyokkal növelt értékének minimumát kell képezni. (A „ $\cdot$ ”-tal jelölt pozíciók nem játszanak szerepet az adott fázis során.)

Mindkét eljárás az inicializálás (ahol a jellemző pontok pozícióján a távolságtérkép kezdőértéke 0, a többi elem esetén pedig  $\infty$ , vagy a tömb típusának megfelelő legnagyobb ábrázolható szám mínusz a legnagyobb súly) után két pásztázó menet jön. A pásztázások során alkalmazott bejárési stratégiák a 4.20. ábrán láthatók. A két bejárás (pásztázás) mindegyike lokális, de nem párhuzamos (tömb)művelet. A lokális függés környezetét és a súlyokat a 4.21. ábra mutatja.

**Algoritmus:**  $(d1, d2, d3)$ -chamfer távolságtérkép számítására

*Input:* az  $A = [a(i, j, k)]$   $n1 \times n2 \times n3$ -as bináris kép

*Output:* a  $B = [b(i, j, k)]$   $n1 \times n2 \times n3$ -as távolságtérkép

// inicializálás

```
for k = 1 to n3 do
  for j = 1 to n2 do
    for i = 1 to n1 do
      if a(i, j, k) = 1 then
        b(i, j, k) = 0
      else
        b(i, j, k) =  $\infty$ 
```

// pásztázás előre

```
for k = 1 to n3 do
  for j = 1 to n2 do
    for i = 1 to n1 do
      b(i, j, k) = min{
        b(i - 1, j - 1, k - 1) + d3, b(i - 1, j - 1, k) + d2, b(i - 1, j - 1, k + 1) + d3,
        b(i - 1, j, k - 1) + d2, b(i - 1, j, k) + d1, b(i - 1, j, k + 1) + d2,
        b(i - 1, j + 1, k - 1) + d3, b(i - 1, j + 1, k) + d2, b(i - 1, j + 1, k + 1) + d3,
        b(i, j - 1, k - 1) + d2, b(i, j - 1, k) + d1, b(i, j - 1, k + 1) + d2,
        b(i, j, k - 1) + d1, b(i, j, k) }
```

// pásztázás hátra

```
for k = n3 downto 1 do
  for j = n2 downto 1 do
    for i = n1 downto 1 do
      b(i, j, k) = min{
        b(i, j, k), b(i, j, k + 1) + d1,
        b(i, j + 1, k - 1) + d2, b(i, j + 1, k) + d1, b(i, j + 1, k + 1) + d2,
        b(i + 1, j - 1, k - 1) + d3, b(i + 1, j - 1, k) + d2, b(i + 1, j - 1, k + 1) + d3,
        b(i + 1, j, k - 1) + d2, b(i + 1, j, k) + d1, b(i + 1, j, k + 1) + d2,
        b(i + 1, j + 1, k - 1) + d3, b(i + 1, j + 1, k) + d2, b(i + 1, j + 1, k + 1) + d3 }
```

A lineáris idejű eljárást a 4.22. ábra segít megérteni.

A  $B$  távolságtérkép minden elemét háromszor látogattuk meg és mindhárom esetben  $\mathcal{O}(1)$  (konstans) időt igényel egy-egy elem új értékének meghatározása. Tehát a fenti program komplexitása valóban  $\mathcal{O}(n)$ , ahol a kép pontjainak száma  $n$ .

Az  $n$ -dimenziós  $j$ -szomszédsági reláción alapuló  $D_j^n$  távolságokon ( $j = 4, 8$ , ha  $n = 2$  és  $j = 6, 18, 26$ , ha  $n = 3$ ) alapuló távolság-transzformációk megkaphatók alkalmas súlyozású

0	0	0	0	.	.	.	.
0	0	1	0	.	.	0	.
0	0	0	0	.	.	.	.
0	0	0	0	.	.	.	.
0	1	0	0	.	0	.	.
0	0	0	0	.	.	.	.

.	.	.	.	7	4	3	4
.	.	0	3	6	3	0	3
.	4	3	4	7	4	3	4
8	7	6	7	4	3	4	7
11	0	3	6	3	0	3	6
4	3	4	7	4	3	4	7

4.22. ábra. Példa chamfer távolságtérkép számítására  $(d1, d2) = (3, 4)$  súlyokkal. A bemenő bináris kép (fent, bal) kettő jellemző pontot tartalmaz. A távolságtranszformáció inicializálása után kapott tömb (fent, jobb), ahol a „.” pozíciók értéke  $\infty$ . A „félkész” távolságtérkép (lent, bal) a pásztázás előre fázis után. A pásztázás hátra fázis végén kialakult távolságtérkép (lent, jobb).

chamfer távolságtérképek kiszámításával:

$$D_4^2 : d1 = 1, d2 = \infty;$$

$$D_8^2 : d1 = 1, d2 = 1;$$

$$D_6^3 : d1 = 1, d2 = \infty, d3 = \infty;$$

$$D_{18}^3 : d1 = 1, d2 = 1, d3 = \infty;$$

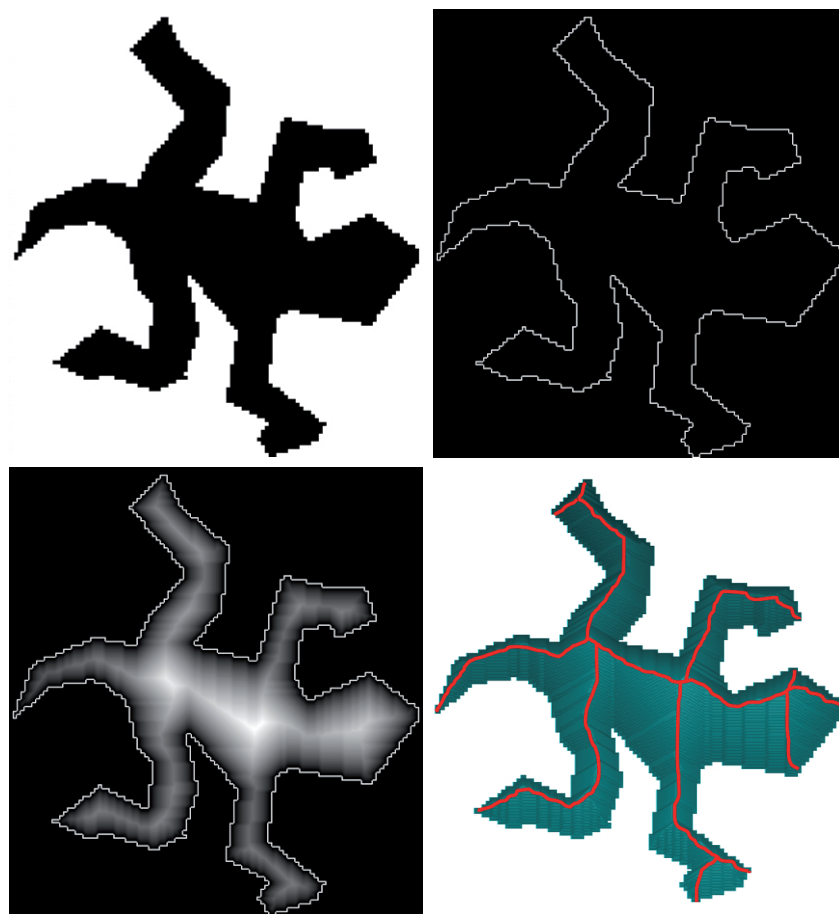
$$D_{26}^3 : d1 = 1, d2 = 1, d3 = 1.$$

Egész súlyok mellett a távolságtérkép elemei is egészek. A fenti módszer célja, hogy az egymástól irracionális ( $\sqrt{2}$  és  $\sqrt{3}$ ) távolságra lévő rácspontok távolságát racionális számokkal közelítsük.

Danielsson hasonló módszert javasolt „közel-euklidészi” távolságtérképre (*almost Euclidean distance map*), ami csak a jellemző pontok néhány speciális konfigurációja esetén nem ad pontos eredményt [21]. Módszere vektor-komponensű (egész számpárokat tároló) tömbbel dolgozik és az inicializálás után minden elemet négyszer látogat meg. Megjegyezzük, hogy számos eljárást javasoltak *hibamentes euklidészi távolságtérkép* (*error-free Euclidean distance map*) előállítására is [20, 79].

A váz távolság-transzformációval történő meghatározása két lépésből áll:

1. A távolságtérkép kiszámítása, ahol a jellemző pontok az input bináris kép háttérpontjai vagy az objektumok határpontjai (ahol a határpontok olyan háttérpontok, amelyeknek van objektumpont szomszédjuk).



4.23. ábra. Távolság-alapú vázkijelölés egy bináris kép (fent, bal) objektumára. A távolságtranszformáció bemenete (fent, jobb), ahol a jellemző pontok az objektum határpontjai. A távolságtérkép (lent, bal), ahol csak az objektumpontokban jelenítettük meg a távolságértékeket (minél nagyobb, annál világosabb). A távolságtérkép domborzatként megjelenítve (lent, jobb), ahol a távolság-értékeknek magasság-értékek felelnek meg. A domborzaton piros vonalak jelölik a hegygerinceket, vagyis a vázat.

2. A vázpontok, vagyis a lokális maximumok „hegygerincek” (ridge) megkeresése a távolságtérképen.

Az módszert a 4.23. ábrával illusztráljuk.

A távolságtérképen a „hegygerincek”, vagyis a vázhoz tartozó pontok kijelölése nehéz és költséges feladat, különösen magasabb,  $d \geq 3$  dimenzióban. A lokális maximumhelyeket kereső módszerek a differenciálgeometria folytonos függvényekre vonatkozó eredményein alapulnak.

Legyen  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  egy folytonos függvény, amelyhez az alábbi ( $d \times d$  méretű) Hesse-féle mátrix tartozik:

$$H(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1} & \frac{\partial^2 f}{\partial x_d \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{pmatrix}.$$

Határozzuk meg a negatív Hesse-féle mátrix ( $-H(f)$ )

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$$

sajátértékeit és az azokhoz tartozó

$$\vec{e}_1, \vec{e}_2, \dots, \vec{e}_d$$

sajátvektorokat, ahol

$$-H(f) \cdot \vec{e}_i = \lambda_i \cdot \vec{e}_i \quad (i = 1, 2, \dots, d).$$

A  $P$  pontban az  $f$  függvénynek  $(d-m)$ -dimenziós ( $m=1, 2, \dots, d$ ) hegygerince található, ha  $f$ -nek lokális maximuma van a  $P$  pontban az  $\vec{e}_1, \vec{e}_2, \dots, \vec{e}_m$  irányokban, vagyis a  $-H(f)$  mátrix legnagyobb  $m$  darab sajátértékéhez tartozó sajátvektorainak mentén. *Ideális hegygerinc* (ideal ridge) esetén az  $f$  függvény csak a hegygerinc merőleges irányban változtatja meg az értékét és konstans a hegygerinc irányában. Ekkor a  $-H(f)$  mátrix rangja  $m$  és  $\lambda_j = 0$  ( $j = m+1, \dots, d$ ).

A gyakorlatban nem egy folytonos  $f$  függvényen, hanem egy diszkrét  $B$  távolságtérképen kell a hegygerinceket megtalálni. Diszkrét esetben nem várható el, hogy ideális hegygerincet találjunk. 2D-ben a távolságtérkép egy eleme hegygerincnek (vázpontnak) felel meg, ha abban a pontban egy „nagy” pozitív és egy „közel nulla” sajátértéke van. A „nagy” pozitív sajátértékhez tartozó sajátvektor adja meg a hegygerinc merőleges irányt. A Hesse-féle mátrix digitális közelítése véges differenciák kiszámításával történhet a távolságtérkép megfelelő ( $\sigma$  paraméterű) Gauss-szűrővel való simítását követően.

Távolság-transzformációval történő vázkijelölésre számos algoritmust javasoltak. A terület iránt mélyebben érdeklődő olvasónak a Siddiqi és Pizer szerkesztette kiváló könyvet ajánljuk a figyelmébe [62].

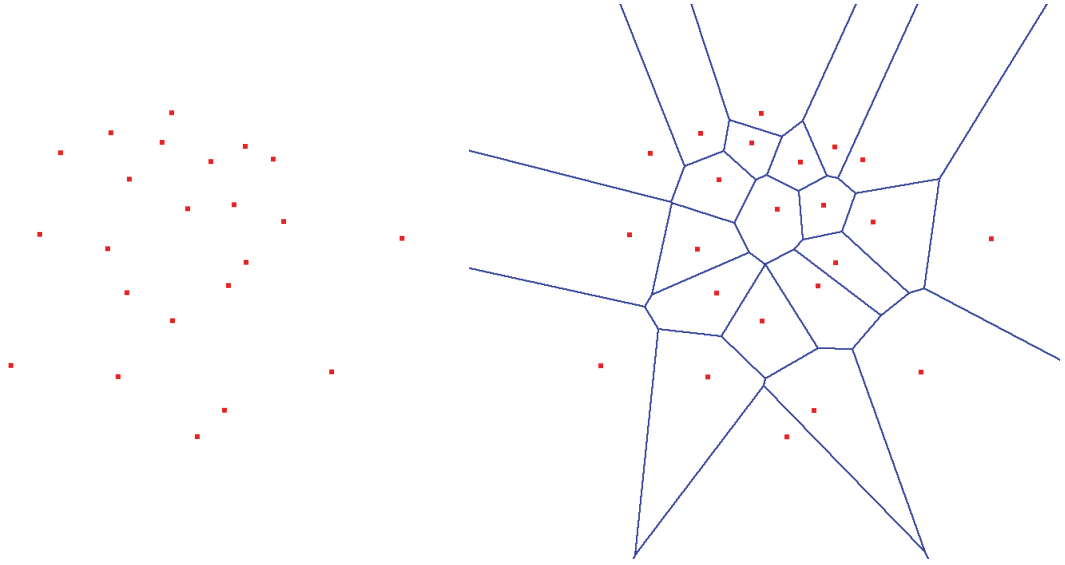
A távolság-transzformáción alapuló eljárások nem garantálnak topológiailag korrekt vázat, ha a távolságtérképen „közel lapos” régiók találhatók. Ugyanakkor a módszer fő erénye az, hogy hibamentes euklidészi távolságtérkép mellett geometriailag korrekt vázat eredményez.

### 4.3. A Voronoi váz

Topológiailag és geometriailag egyaránt korrekt váz meghatározására ígéretes a *Voronoi-diagramon* [3, 49, 76] alapuló módszer.

Először a 2-dimenziós Voronoi-diagramot mutatjuk be. Legyen  $G = \{g_1, \dots, g_t\}$   $t$  (különböző) diszkrét pontot tartalmazó halmaz, ahol  $g_i \in \mathbb{Z}^2$  ( $i = 1, \dots, t$ ). A Voronoi-diagram





4.24. ábra. Egy generálómalmaz (bal) Voronoi-diagramja (jobb).

0-, 1- és 2-dimenziós elemekből áll, melyeket rendre *Voronoi-csúcsoknak*, *Voronoi-éleknek* és *Voronoi-régióknak* neveznek.

A pontosan  $t$  darab Voronoi-régió mindegyike a  $G$  generálómalmaz egy-egy eleméhez rendelhető:

$$R_i(G) = \{ p \mid \forall j : 1 \leq j \leq t, j \neq i \quad D(p, g_i) < D(p, g_j) \}$$

( $i = 1, \dots, t$ ), ahol „ $D$ ” az euklidészi távolságot jelöli.

Egy régióba azon pontok kerülnek, amelyek közelebb vannak a generálómalmaz egy adott eleméhez, mint az összes többihez. Könnyen belátható, hogy valamennyi  $R_i(G)$  Voronoi-régió konvex (egy régió bármely kettő pontjára a kiválasztott pontokat összekötő szakasz egésze is a régióba esik), egyértelműen létezik és nem üres (mivel  $g_i \in R_i(G)$ ). Bizonyos régiók korlátosak, míg mások területe végtelen is lehet.

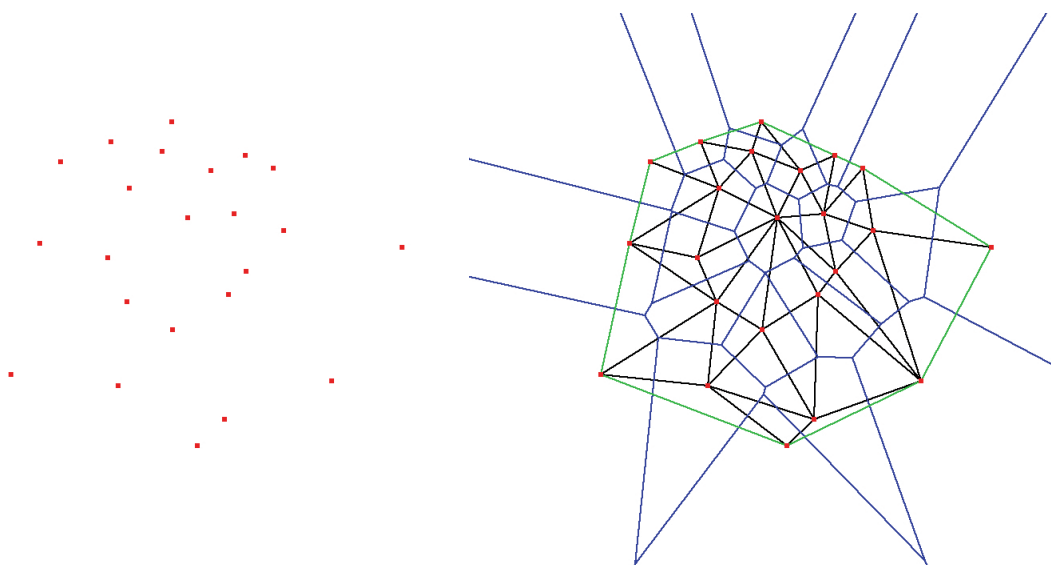
A Voronoi-éleket pontosan kettő  $G$ -beli ponttól egyenlő távolságra és minden más generálóponttól távolabbra eső pontok alkotják:

$$E_{ij}(G) = \{ p \mid D(p, g_i) = D(p, g_j) \text{ és} \\ \forall k : 1 \leq k \leq t, k \neq i, k \neq j \quad D(p, g_i) < D(p, g_k) \}$$

( $1 \leq i \neq j \leq t$ ). Megjegyzendő, hogy bizonyos  $g_i$  és  $g_j$  pontokra  $E_{ij}$  üres is lehet, ugyanakkor bizonyos Voronoi-élek lehetnek végtelen hosszúságúak is. Belátható, hogy  $E_{ij}$  része a  $g_i$  és a  $g_j$  pontokat összekötő szakasz szakaszfelező merőleges egyenesének.

Egy Voronoi-csúcs (legalább) három  $G$ -beli ponttól egyenlő távolságra és a generálómalmaz minden más pontjától távolabbra esik:

$$V_{ijk}(G) = \{ p \mid D(p, g_i) = D(p, g_j) = D(p, g_k) \text{ és} \\ \forall l : 1 \leq l \leq t, l \neq i, l \neq j, l \neq k \quad D(p, g_i) \leq D(p, g_l) \}$$



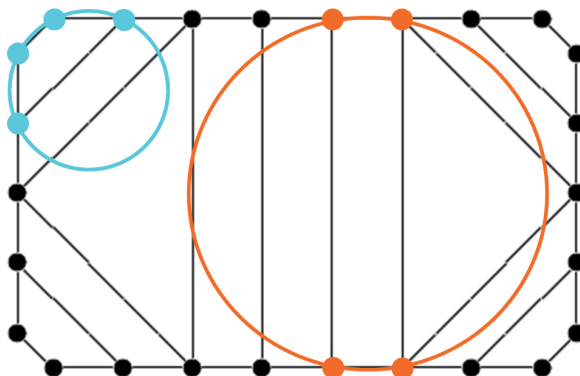
4.25. ábra. Egy generálópontok halmaza (bal) Voronoi-diagramja és a vele duális Delaunay felbontása (jobb).

( $1 \leq i \neq j \neq k \leq t$ ). Megjegyezzük, hogy egyrészt bizonyos  $g_i$ ,  $g_j$  és  $g_k$  pontokra nem létezik a  $V_{ijk}(G)$  csúcs, másrészt fennállhat az is, hogy  $V_{ijk} = V_{ijl}$  ( $k \neq l$ ), mivel egy csúcstól háromnál több  $G$ -beli pont is eshet azonos (és minimális) távolságra.

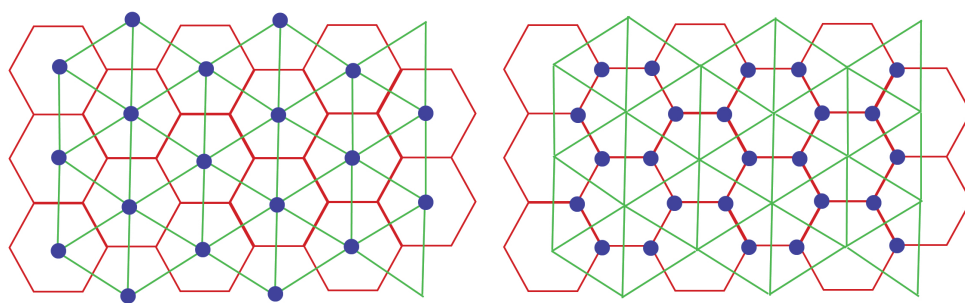
Voronoi-diagramra a 4.24. ábra ad példát.

A  $G$  generálópontokkal adott Voronoi-diagrammal szoros kapcsolatban áll  $G$  Delaunay-felbontása [49, 76], mely  $G$  konvex burkát (*convex hull*) osztja fel háromszögekre. Egy 2-dimenziós diszkrét pontfelhalmaz konvex burka az a legkisebb területű konvex poligon, ami tartalmazza a pontfelhalmaz valamennyi pontját. (Megjegyezzük, hogy a konvex burok egyértelműen létezik tetszőleges pontfelhalmazra.)

A Delaunay-felbontás a konvex burok egy speciális triangularizációja: a háromszögek köré írt körlapoknak nem belső pontja a generálópontok egyetlen pontja sem (ld. 4.25. ábra). Megjegyezzük, hogy a Delaunay-triangularizáció nem egyértelmű minden esetben, mivel egy (lehetséges) felbontásban ugyanaz lehet a köréírt köre egyszerre több háromszögnek is. Az ilyen háromszögek olyan konvex poligont alkotnak, amelynek tetszőleges triangularizáció-



4.26. ábra. Példa Delaunay felbontásra, ahol háromszögeken kívül négyszögek is előfordulnak.

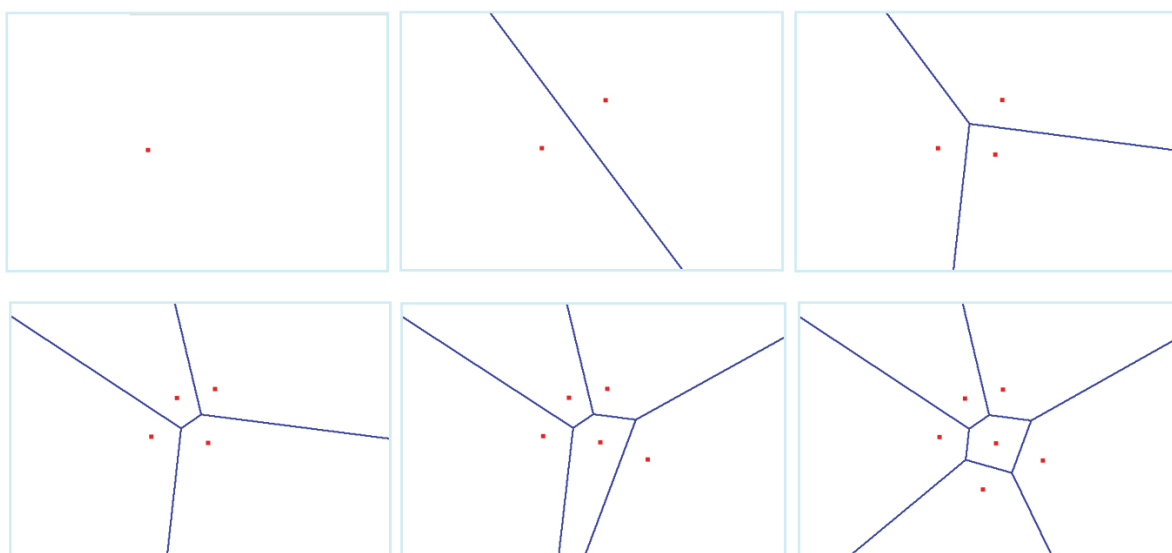


4.27. ábra. A hatszögmozaik és a vele duális háromszögmozaik. Amennyiben a háromszögek csúcspontjai (a hatszögek középpontjai) a generálóhalmaz elemei, úgy a Voronoi-diagram régiói a hatszögek, a Delaunay felbontás 2D elemei pedig a háromszögek (bal). Ha a generálóhalmaz elemeit a háromszögek középpontjaiba (a hatszögek csúcsaiba) helyezzük, úgy a Voronoi-diagram régiói a háromszögek lesznek, a Delaunay felbontás 2D elemei pedig a hatszögek, melyeknek már tetszőleges háromszögekre bontása korrekt triangularizációt eredményez (jobb).

jára teljesül a Delaunay–felbontás kritériuma. A 4.26. ábra példáján a Delaunay–felbontásban négyszögek is előfordulnak, melyeknek bármelyik átlója szerepelhet a triangularizációban.

A Delaunay–felbontásban is 0-, 1- és 2-dimenziós elemek szerepelnek, amelyeket rendre *Delaunay–csúcsoknak*, *Delaunay–éleknek* és *Delaunay–régióknak* neveznek. A Delaunay–csúcsok halmaza maga a  $G$  generálóhalmaz, a Delaunay–éleket a felbontás háromszögeinek oldalai alkotják, a Delaunay–régiók pedig a háromszögek.

A Voronoi–diagram és a Delaunay–felbontás egymásnak *duálisa*: az egyik  $d$ -dimenziós ( $d = 0, 1, 2$ ) elemeinek egy-egyértelmű módon megfeleltethetők a másik  $(2 - d)$ -dimenziós elemei. A  $V_{ijk}(G)$  (0D) Voronoi–csúcsnak az a (2D) Delaunay–háromszög felel meg, amely-



4.28. ábra. A Voronoi-diagram meghatározása pont-inkrementációs technikával. A generálóhalmaz elemeit egyenként szűrjük be és az addig elkészült diagramon eszközöljük a „szükséges” módosításokat.

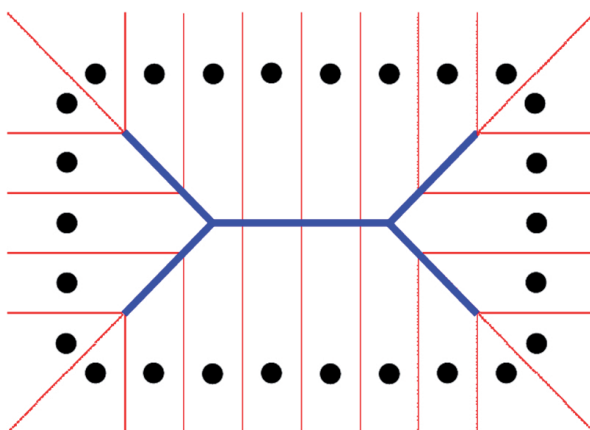
nek csúcsai  $g_i$ ,  $g_j$  és  $g_k$  (a háromszög köré írt kör középpontja éppen  $V_{ijk}(G)$ ). Az  $E_{ij}(G)$  (1D) Voronoi-élhez a  $g_i$  és a  $g_j$  pontokat összekötő (1D) Delaunay-él tartozik. A  $R_i(G)$  (2D) Voronoi-régiónak pedig a  $g_i$  (0D) Delaunay-csúcs felel meg. A 4.27. ábra a duális rácsok/mozaikok és a tárgyalt felbontásaink kapcsolatát illusztrálja.

A 3-dimenziós pontthalmazokra is megadható a Voronoi-diagram és a Delaunay-felbontás. Ekkor a Voronoi-régiók 3D konvex poliéderek lesznek, melyek lapjai, élei és csúcsai adják rendre a diagram 2D, 1D és 0D elemeit. A Delaunay-felbontásban a 3D elemek tetraéderek, melyek (2D) lapjai, (1D) élei és (0D) csúcsai alkotják a felbontás 2D, 1D és 0D elemeit. A dualitás ezúttal is fenáll: az egyik konstrukció  $d$ -dimenziós ( $d = 0, 1, 2, 3$ ) elemeihez a másik  $(3 - d)$ -dimenziós elemei tartoznak.

A 2-dimenziós Voronoi-diagram meghatározására és a Delaunay-felbontásra számos módszert dolgoztak ki, melyek komplexitása  $t$  pontra 2-dimenzióban  $\mathcal{O}(t \cdot \log_2 t)$  és 3-dimenzióban  $\mathcal{O}(t^2)$  [2, 45]. A Voronoi-diagram pont-inkrementációs módszerrel való meghatározását a 4.28. ábra illusztrálja.

A Voronoi-diagram kapcsolatban áll a vázzal: ha a generálóhalmaz pontjait egyenletes, minden határon túl finomodó mintavételezés szerint vesszük az objektum határáról, akkor a Voronoi-diagramnak az objektumba eső elemei a vázhoz konvergálnak [13].

A 4.29. ábra a Voronoi-vázra mutat példát. A példaobjektum – ezúttal is – egy szabályos téglalap, így a Voronoi-diagram összevethető a 4.1. ábrán látható folytonos vázzal.



4.29. ábra. Egy téglalap Voronoi váza az objektum belsejébe eső Voronoi-élekből áll.

A Voronoi-diagram (éppúgy, mint a Delaunay-felbontás) folytonos modell. Diszkrét objektumok esetén a diszkrét generálóhalmaz kijelölése után modellt kell váltani: a folytonos Voronoi-diagramból kell kiválasztani azokat a folytonos elemeket, amelyek a vázhoz tartoznak. Ezért a Voronoi-diagramon alapuló vázkijelölést *félíg folytonos (semi-continuous)* megközelítésnek is nevezik.

A Voronoi-váz meghatározásakor két kérdésre kell választ találni: mely pontokból álljon a generálóhalmaz, és a Voronoi-diagram mely elemei tekintendők relevánsnak, a vázmodellhez tartozónak.

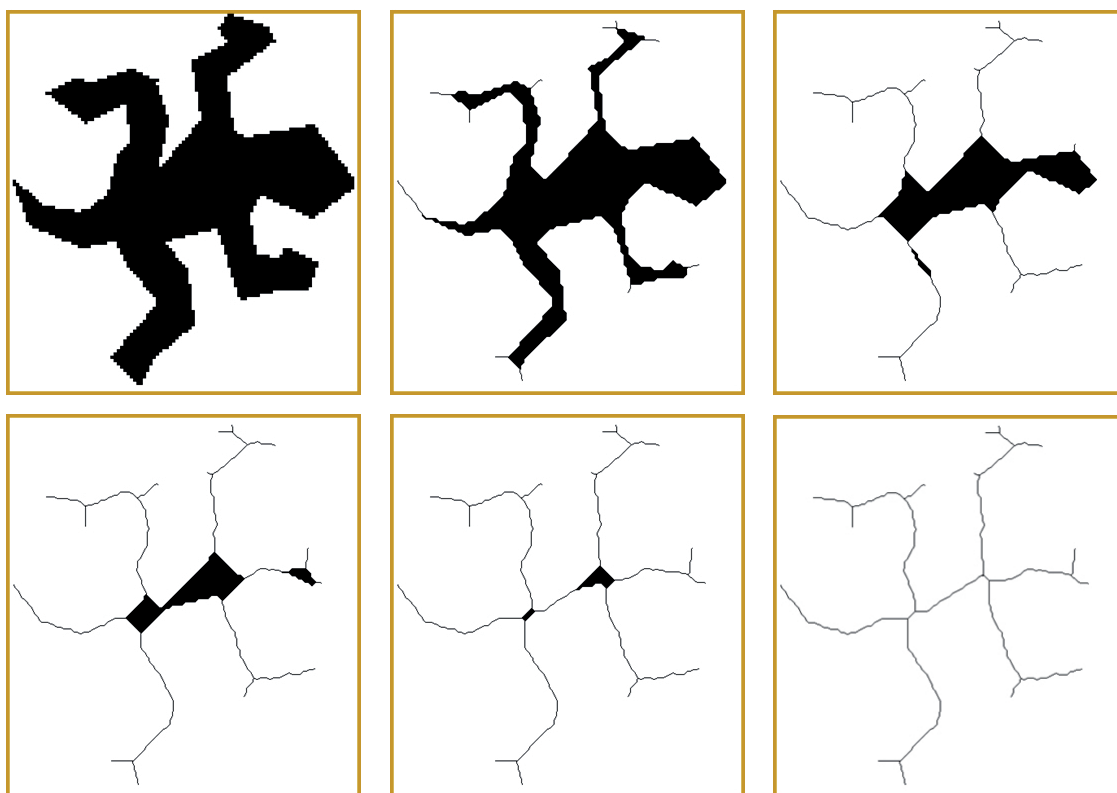
Az első kérdésre egy lehetséges válasz: egy 2-dimenziós képen alkossák a generálóhalmazt a fekete és fehér pixelek (egységnégyzetek) közös oldalainak felezőpontjai, 3-dimenzióban pedig a fekete és fehér egységkockák közös lapjainak középpontjai [45].

A második probléma megválaszolására sokféle szubjektív és heurisztikus módszert tesz lehetővé az a tény, hogy a Voronoi–diagram (vagy a Delaunay–felbontás) összetett adatstruktúrákban tárolható. Ennélfogva a modellek mint folytonos  $0D$ –,  $1D$ –, ...,  $dD$ –elemek halmazai különféle *fontossági mértékek* (*importance measure*) [2, 59] alapján egyszerűsíthetők, tisztíthatók.

A Voronoi–diagramból geometriailag és topológiailag egyaránt korrekt váz vonható ki, aminek utófeldolgozására, szimbólikus leírására számos lehetőség adódik [45, 59]. A módszer hátránya az, hogy a folytonos modell feldolgozása nagymennyiségű, lebegőpontos aritmetikát igénylő számítást igényel.

## 4.4. A vékonyítás

Diszkrét objektumok vázának közelítésére gyakran használják a front–terjedést modellező iteratív eljárást, a vékonyítást. A front–terjedés a diszkrét térben diszkrét folyamat: egy időegység alatt a tűz frontja a tőle „egységnyi” távolságra lévő rácspontokat éri el. A vékonyítás egy iterációs lépése tehát a front–terjedés egy időegységének felel meg. Az eljárás akkor ér véget, ha már nincs „éghető anyag” az objektumon. A vékonyítás menetét a 4.30. ábrán mutatjuk be.



4.30. ábra. Példa vékonyításra. A képek sorfolytonosan: kiindulási objektum, az eredmények rendre 15, 30, 45 és 60 iterációs lépés után és a végeredmény.

A vékonyítás természeténél fogva párhuzamos, mivel a prérítűz analógia szerint a határ valamennyi pontja egyidejűleg kap lánggra, továbbá a tűzfrontok minden irányba egyenletes sebességgel terjednek. Ennélfogva a létező vékonyító eljárások többsége párhuzamos, vagyis az eljárás egy fázisában a törölhetőnek minősülő pontokat egyidejűleg távolítják el [26, 36, 47, 66]. Megjegyezzük, hogy javasoltak szekvenciális vékonyító algoritmusokat is, melyek a kontúrkövetés technikáját alkalmazva imitálják a külső réteg pontjainak egyidejű eltávolítását [36, 66]. A front-terjedés nem csak párhuzamos, hanem lokális is. Ezen második fontos tulajdonság szerint az, hogy egy objektumpont a „tűz áldozata” lesz-e az eljárás következő fázisában vagy sem, eldönthető a kérdéses pont egy – az objektum nagyságától független méretű – szűk környezete alapján.

A vékonyító algoritmusok egy iterációs lépése tehát egy olyan lokális és párhuzamos redukcióval adható meg, ami csak határpontokat töröl. A teljes eljárás ilyen képműveletek sorozatából áll. A vékonyítás menete az alábbi általános sémával írható le:

```
repeat
    a „törölhető” (határ)pontok eltávolítása
until nem történt változás
```

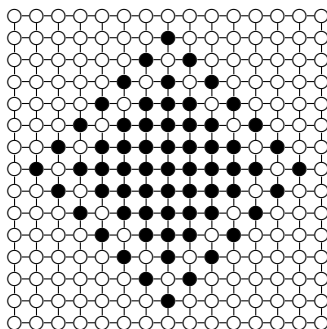
A repeat-ciklus magja a vékonyítás egy iterációs lépésének felel meg. Az egyes vékonyító eljárások az iterációs lépések megszervezésében és a „törölhető” pontokra adott feltételekben különböznek egymástól.

A vékonyítás a vázzal szemben támasztott topológiai kritérium teljesítésére helyezi a hangsúlyt, vagyis topológia-megőrző redukciós képműveletnek kell lennie. A másik, a geometriai kritériumnak viszont nem tud maradéktalanul eleget tenni, mivel az adott digitális térben ( $\mathbb{Z}^2$ -ben vagy  $\mathbb{Z}^3$ -ben) alkalmazott szomszédsági relációk által indukált „távolságok” nem közelítik „jól” az euklidészi távolságot. A vékonyításon és a távolság-transzformáción alapuló vázkijelölés egymást kiegészítő módszerek: ami az egyiknél erény, az a másiknál hiányosság. Topológiaiilag és geometriaiilag egyaránt korrekt váz meghatározása lehetséges a fenti két módszer együttes alkalmazásával, ahol a távolság-transzformációval meghatározott vázat „horgonyként” kezeli az azt követő vékonyító eljárás [55, 67]. A kombinált módszernél a távolság-transzformáció ad egy, a geometriai kritériumnak eleget tevő, de nem összefüggő „félkész” vázat, amit a vékonyítás megőriz és kiegészít úgy, hogy az a topológiai feltételt is teljesítse.

Sokféle topológia-megőrző lokális és párhuzamos redukció létezik, többségük nyilvánvalóan nem mutat hasonlóságot a front-terjedéssel. Vékonyító algoritmusok tervezésekor tehát nem csak arra kell ügyelni, hogy úgy töröljünk, hogy az eredetivel topológiaiilag ekvivalens képet kapjunk. Arra is figyelni kell, hogy mikor ne töröljünk, vagyis azok a pontok, ahol a tűzfrontok találkoznak, maradjanak meg akkor is, ha elhagyásuk megőrizné a topológiát. Ezeket a védett pontokat *végpontoknak* nevezzük.

A 3D vékonyító algoritmusokat az általuk kivont közelítő váz jellege szerint két osztályba soroljuk: vannak, amelyek a *középvonalra* vékonyítanak (*medial line thinning*), mások pedig *középfelületre* (*medial surface thinning*). Amennyiben nem alkalmazunk végpont-kritériumot, úgy a vékonyítás az objektum *topológiai magját* (*topological kernel*) határozza meg. Ez esetben a vékonyítás hatása ugyanaz, mint a csak redukcióval élő *zsugorításnak* (*shrinking*).





4.31. ábra. Példa kiterjedt, de nem-vékonyítható objektumra. A (8-összefüggő) objektum egyetlen fekete pontja sem egyszerű. Bármelyik fekete pont törlése vadonatúj (4-összefüggő) üreget hozna létre, (egy vagy kettő) üreget szüntetne meg vagy (kettő vagy három) üreget olvasztana össze.

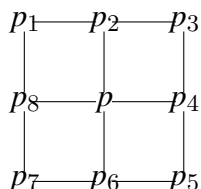
Az objektumok iteratív „hámozására” számos eljárást javasoltak, így megfogalmazták az alábbi feltételeket [52, 70]. Egy eljárásnak kijár a ”vékonyító algoritmus” minősítés, ha

1. Topológia-megőrző.
2. 2-dimenziós esetben egy pont vastag (1D) görbéket, 3-dimenzióban pedig (2D) felszín-szegmenseket változatlanul hagy.
3. 2-dimenziós esetben megváltoztatja a  $2 \times 2$ -es négyzet-alakú objektumot, illetve 3-dimenzióban pedig a  $2 \times 2 \times 2$ -es kocka-alakú objektumot.

Az 1. feltétel alapkövetelmény, a 2. azt írja elő, hogy a vékonyítás különbözzön a zsugorítástól (mikor ne töröljünk tovább), míg a 3. feltétel (nem a törlést korlátozza, hanem) a minél „vékonyabb” objektumok elérését próbálja elősegíteni.

Érdekességgéppen megjegyezzük, hogy a 4.31. ábrán látható objektum nem vékonyítható, mivel nem tartalmaz a topológia megőrzése mellett törölhető pontot [66]. Hasonló példaobjektumok – tetszőleges méretben – 3-dimenzióban is konstruálhatók.

A vékonyítás módszerének felvázolását két algoritmus (egy 2-dimenziós és egy 3-dimenziós) bemutatásával zárjuk. A Rosenfeld által javasolt eljárás [52] minden egyes iterációs lépést négy egymást követő al-iterációra bont fel. Az eljárás pszeudokódja:



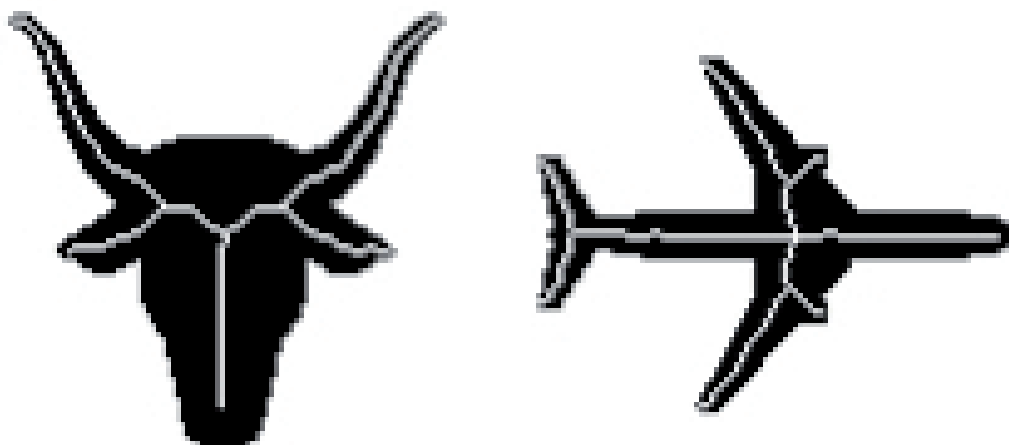
4.32. ábra. A  $p$  pont 8-szomszédainak indexelése.

**Algoritmus:** Rosenfeld (2D)

```

repeat
  for  $i = 2, 4, 6, 8$  do
    egyidejű törlése mindazon  $p = 1$  pontoknak, melyekre
       $p_i = 0$ ,
       $|p_2 - p_1| + |p_3 - p_2| + \dots + |p_8 - p_7| + |p_1 - p_8| = 2$  és
       $p$ -nek legalább kettő darab 8-szomszédja objektumpont
until nem történt változás

```



4.33. ábra. Példák Rosenfeld algoritmusával kapott középvonalakra. A két tesztobjektumra rávetítettük a vázközelítéseket. Megállapíthatjuk, hogy az eljárás a „közel szimmetrikus” objektumokra „közel szimmetrikus” középvonalakat eredményezett.

A lokális (és párhuzamos) eljárásban egy pont törölhetősége annak 8-szomszédsága ( $3 \times 3$ -as környezete) alapján dönthető el. Egy  $p$  pont 8-szomszédait a 4.32. ábra szerint indexeltük.

Az algoritmusra a 4.33. ábra mutat példát.

3-dimenziós vékonyításra egy szekvenciális algoritmust vázolunk fel, melyhez szükségünk van az *egyszerű pont* (*simple point*) [35] fogalomra. Egy objektumpont egyszerű, ha törlése topológia-megőrző redukció. A leggyakrabban használt szomszédsági relációk mellett az egyszerűség lokális tulajdonság: 2-dimenzióban eldönthető a kérdéses pont  $3 \times 3$ -as, 3-dimenziós esetben pedig a  $3 \times 3 \times 3$ -as környezete alapján. A Palágyi és munkatársai által (a csőszerű objektumok középvonalának meghatározására) javasolt algoritmus [48] pszeudokódja (a végpontok újraellenőrzése nélkül):

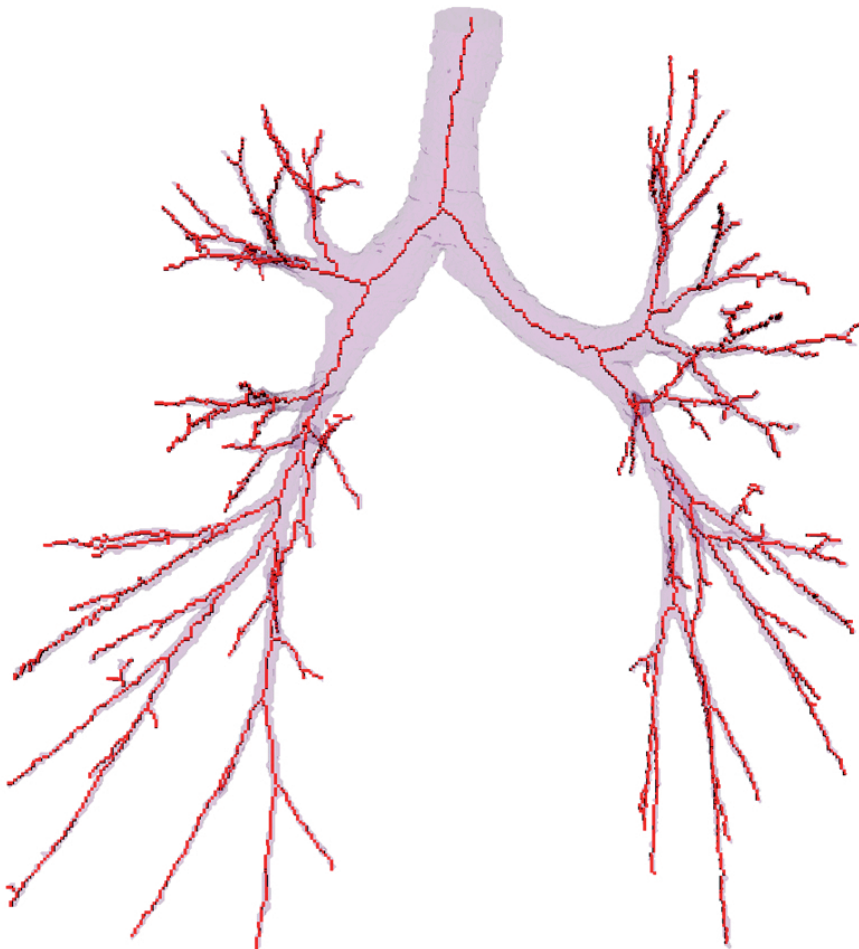
**Algoritmus:** Palágyi és munkatársai (3D)

```

repeat
  for  $i = U, D, N, E, S, W$  do
    jelöljük meg törlésre mindazon  $p = 1$  pontokat, melyekre
     $p$  határpont az  $i$ -irány szerint
    (vagyis az  $i$ -irányba eső 6-szomszédja 0),
     $p$  egyszerű pont és
     $p$ -nek legalább kettő darab 26-szomszédja objektumpont
    (azaz  $p$  nem végpont)
    for minden megjelölt  $p$  pontra do
      if  $p$  egyszerű és nem végpont az aktuális képen then
        töröljük a  $p$  pontot a képről
until nem történt változás

```

A szekvenciális algoritmusra a 4.34. ábra mutat példát.



4.34. ábra. Egy 3-dimenziós CT-vizsgálatból szegmentált emberi légútfa és annak vékonyítással meghatározott középvonala.

## 5. fejezet

# Fejlett szegmentációs módszerek

A jelen fejezetben négy fejlett szegmentációs technikát tekintünk át.

Az első módszer a deformálható határokon, az evolúciós kontúrokon alapul. Két ilyen szegmentációs technikát mutatunk be, az aktív kontúr és a level set módszert.

A másodikat, a vízválasztó szegmentálást egy földrajzi analógia ihlette: a 2-dimenziós többszintű képek felszíneknek, domborzatoknak tekinthetők, ha a képpontok intenzitásainak magasság-értékek felelnek meg. Egy domborzat vízgyűjtő medencéi a kép szegmensei, a vízválasztó vonalak pedig a szegmens-határok.

A harmadik technika a chamfer illesztés, mely a mintaillesztéssel történő képszegmentáció egy megközelítése. Az illesztés „jóságát” megadó mértéket ekkor chamfer távolságtérképekből (ld. 3. fejezet) származtatjuk.

A negyedik módszer a képfeldolgozásban is egyre növekvő fontosságú fuzzy-megközelítést használja. Az alfejezetben a Udupa és Herman által bevezetett fuzzy-összefüggőségen alapuló objektum-meghatározást mutatjuk be.

### 5.1. Szegmentálás deformálható határokkal

Az alfejezetben a két alapvető deformálható határokon alapuló technikát mutatunk be.

#### 5.1.1. Aktív kontúr

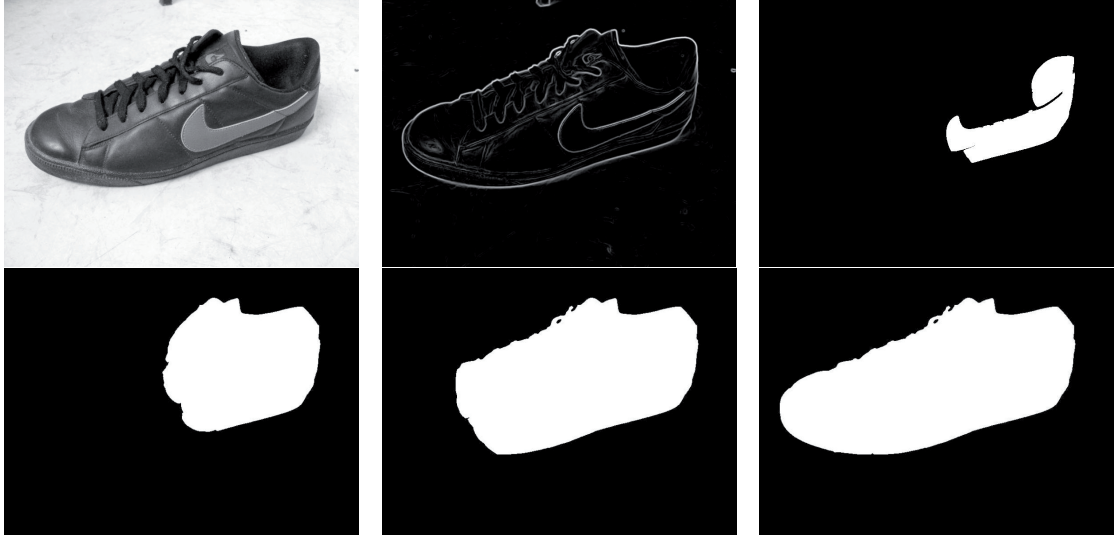
Az *aktív kontúr* (*active contour; snake*) modellt Kass, Witkin és Terzopoulos vezették be [34]. A modellt 3-dimenziós esetben *aktív felszínnek* (*active surface*) említik az irodalomban.

Az aktív kontúr egy  $v(s) = (s(x), s(y))$  parametrikus görbe, melynek az alakját és a mozgását egy olyan  $E_{snake}$  energiafüggvény befolyásolja, mely akkor veszi fel a minimumát, ha a görbe „rásimul” a szegmentálandó objektum határára.

Kass, Witkin és Terzopoulos az energiafüggvényt a következő alakban adta meg:

$$E_{snake} = \int_0^1 (E_{int}(v(s)) + E_{ext}(v(s)) + E_{image}(v(s))) ds,$$

ahol



5.1. ábra. Példa aktív kontúrral történő szegmentációra. A képek sorfolytonosan rendre: kiindulási kép, gradiens kép, kiindulási szegmens, a görbe-evolúció két közbűlső eredménye és a végeredmény.

- az  $E_{int}(v(s))$  a belső energia-komponens, melyre:

$$E_{int}(v(s)) = \frac{1}{2} \left( \alpha(s) \cdot |v_s(s)|^2 + \beta(s) \cdot |v_{ss}(s)|^2 \right),$$

ahol a  $v_s(s)$  első derivált felelős a görbe nyúlékonyságáért/rugalmasságáért, a  $v_{ss}(s)$  második derivált pedig annak hajlékonyságát/merevségét szabályozza, míg az  $\alpha(s)$  és a  $\beta(s)$  szabályozza a két tag relatív fontosságát;

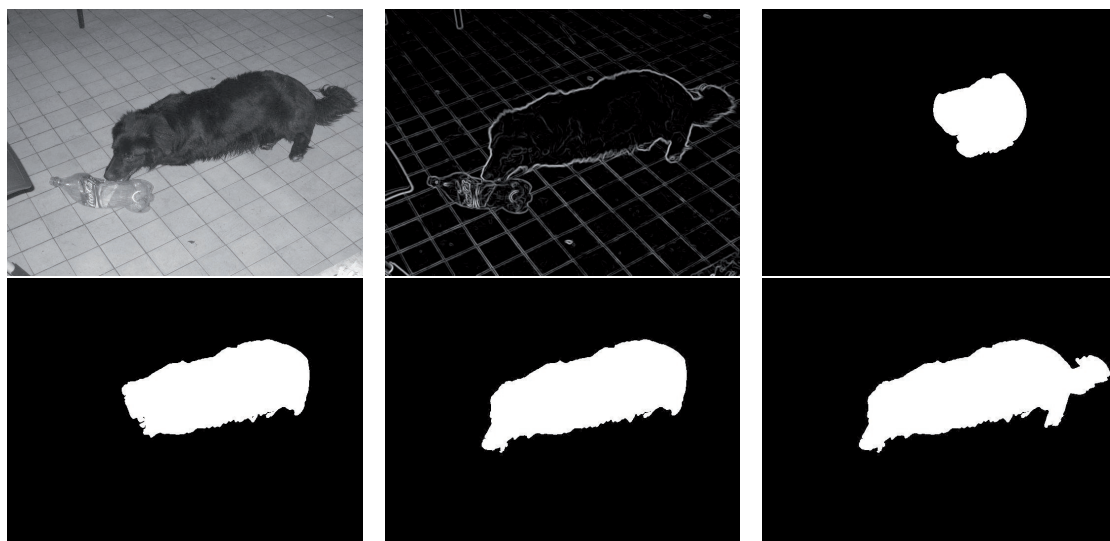
- az  $E_{ext}(v(s))$  a külső komponens, melynek az a szerepe, hogy a határ felé „terelje” a görbét;
- az  $E_{image}(v(s))$  komponens az  $I(x, y)$  szegmentálandó képtől függ, három függvény súlyozott/lineáris kombinációja:

$$E_{image}(v(s)) = w_{line} \cdot E_{line} + w_{edge} \cdot E_{edge} + w_{term} \cdot E_{term},$$

ahol  $E_{line} = I(x, y)$  a legegyszerűbb, ugyanakkor alkalmas választás, az  $E_{edge}$  komponensnek érdemes a  $-(\nabla^2 G_\sigma * I)^2$  függvényt venni (ahol a képünk egy Gauss függvény Laplace transzformáltjával való konvolúciójával történik az éldetektálás), míg az  $E_{term}$  termináló komponens, amit a kép  $C(x, y) = G_\sigma(x, y) * I(x, y)$  „enyhén” simított verziójából képzünk az alábbi formula szerint:

$$E_{term} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{\sqrt{(C_x^2 + C_y^2)^3}}.$$

Sajnos terjedelmi okok miatt a numerikus implementációt nem áll módunkban itt taglalni, viszont megemlítjük a módszert hatékonyabbá tevő külső erőteret, a *gradiens vektor mezőt*



5.2. ábra. Példa aktív kontúrral történő szegmentációra. A képek sorfolytonosan rendre: kiindulási kép, gradiens kép, kiindulási szegmens, a görbe-evolúció két közbűlső eredménye és a végeredmény.

(*gradient vector field, GVF*), mely segít a görbét a korrekt határhoz terelni még konkáv objektumokra is. Az  $I(x, y)$  képre az  $(u_1(x, y), u_2(x, y))$  GDF az alábbi (Euler) egyenletek megoldásával határozható meg:

$$\begin{aligned}\mu \cdot \nabla^2 u_1 - (u - I_x) \cdot (I_x^2 + I_y^2) &= 0 \\ \mu \cdot \nabla^2 u_2 - (u - I_x) \cdot (I_x^2 + I_y^2) &= 0,\end{aligned}$$

ahol a  $\mu$  együtthatót zajos képek esetén érdemes nagyobbra választani. A GDF tehát egy olyan kép, melynek az elemei kételemű vektorok, melyek a legközelebbi él felé mutatnak.

Az aktív kontúrral történő szegmentációra két példával szolgálunk az 5.1. és az 5.2. ábrákkal.

### 5.1.2. Level set

Az aktív kontúr modell rovására írják azt, hogy nehezen kezeli az evolúció során az önmagát átmetsző görbéket és a topológiai változásokat. A fentiek kiküszöbölésére Osher és Sethian a *level set* módszert, numerikus technikát javasolta a határ és az alak követésére [46]. A megközelítés előnye, hogy az objektumok parametrizálása nélkül képes rögzített képrácson dolgozni és kezelni tudja a topológiai változásokat is. A level set módszer a szegmentáláson kívül alkalmas többfázisú mozgáskövetésre és regisztrációra is.

A level set függvények a problémát eggyel megnövelt dimenzióba viszik át, vagyis 2-dimenziós esetben a  $z = \varphi(x(t), y(t), t)$  függvény változik (emelkedik, esik, szűkül, tágul). A  $t=0$  időpillanatban a függvény (előjeles távolságfüggvény) negatív értékeket vesz fel a kezdeti kontúron kívül, nulla a kontúron és pozitív azon belül. A kontúr a 3-dimenziós függvény felszínének a  $z=0$  szinten lévő keresztmetszete (mely a level set függvénnyel együtt változik).





5.3. ábra. Példa level set szegmentációra. A kiindulási kép (bal) és topológiai változásokat is kezelő szegmentációja (jobb).

A level set felszín mozgatásához meg kell adni egy alkalmazás-specifikus  $F$  sebesség-függvényt (*velocity/speed function*), mely pl. időtől, pozíciótól, görbülettől is függhet. Az eljárás lépései:

1. Definiáljuk az  $F$  sebesség-függvényt.
2. Építsük fel a level set függvény  $\varphi(x(0), y(0), 0)$  kezdőértékét a kezdeti kontúr (zárt görbe) alapján.
3. Igazítsuk a  $\varphi$  level set függvényt a  $t$  idő függvényében (ahol  $\varphi(x(t), y(t), t) = 0$  az aktuális kontúr). A (diszkrét rácson numerikusan) megoldandó Hamilton-Jacobi egyenlet:

$$\frac{\partial \varphi}{\partial t} = -F \cdot |\nabla \varphi| = -F \cdot \sqrt{\frac{\partial \varphi^2}{\partial x} + \frac{\partial \varphi^2}{\partial y}}.$$

A gradiens-alapú  $F$  sebesség-függvényt a következő alakban adják meg:

$$F = 1 - \varepsilon \cdot \kappa + \beta \cdot (\nabla \varphi \cdot \nabla |\nabla I|),$$

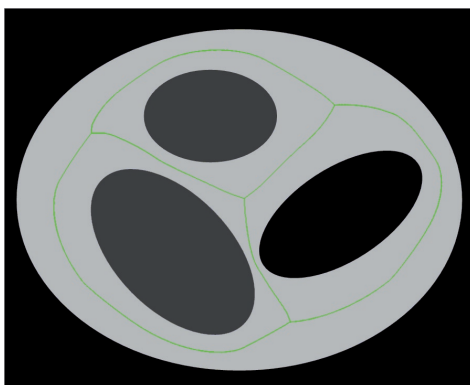
ahol a  $+1$  biztosítja a belülről történő felfújást, az  $\varepsilon \cdot \kappa$  tag (melyben  $\kappa$  jelöli a görbületet) csökkenti a kontúr görbületét és az utolsó tag (melyben  $I$  a szegmentálandó kép) megállítja a felfújást a határon.

Terjedelmi korlátok miatt további részletek ismertetésére és a numerikus megoldás bemutatására nem vállalkozhatunk. A területbe elmélyülni kívánók figyelmét két Sethian-műre hívjuk fel: az első a *fast marching level set* módszerről szól [57], a másik pedig egy level set módszerbe bevezető könyv [58].

A level set szegmentációra az 5.3. ábra ad példát.

## 5.2. Vízválasztó szegmentálás

A vízválasztó szegmentálás (*watershed segmentation*) a régiónövelést és az él-detektálást kombinálja. Csoportosítja a képpontokat a régió minimuma körül és a szomszédos régiók

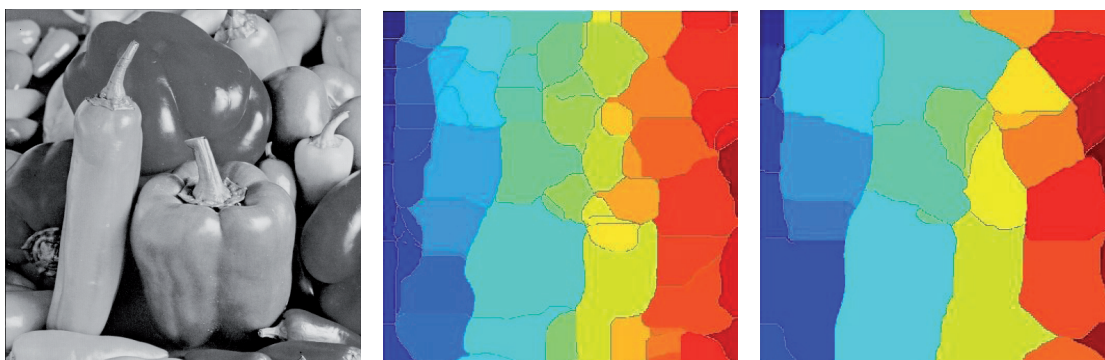


5.4. ábra. Egy többszintű képen Vincent és Soille vízválasztó szegmentáló algoritmus a zöld régió-határokat eredményezi.

közötti határvonalak megfelelnek a vízválasztóknak, a régiók belső pontjai pedig a vízgyűjtő területeket reprezentálják.

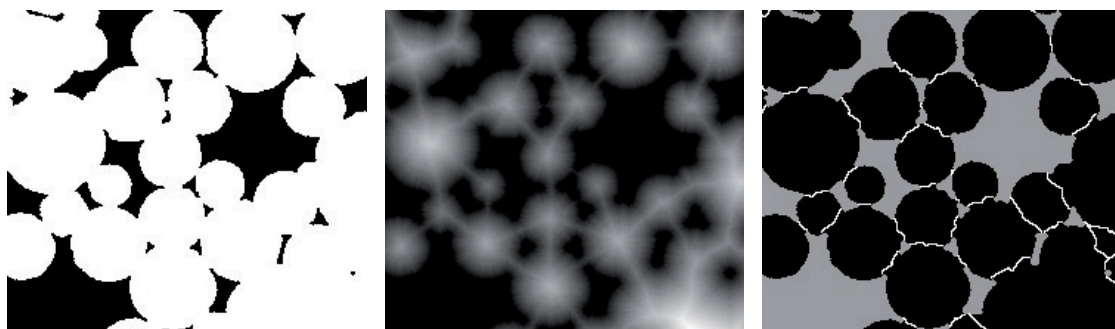
A vízválasztó szegmentálás Vincent és Soille által javasolt klasszikus módszere [74] a kép lokális minimumaiból (mint a medencék aljáról) kiindulva szimulált árasztással hízlalják a vízgyűjtő területeket. Gátakat emelnek oda, ahol kettő vagy több medence-fenékről induló vizek összeérnek és végül a gátak rendszere adja a régiók határát. A Vincent és Soille vízválasztó szegmentálásra egy egyszerű példán az 5.4. ábra mutatja be.

A vízválasztó szegmentálás módszerei általában túlszegmentálják a képet, vagyis túl sok kisméretű régiót eredményeznek, így a vízválasztó szegmentálást a *régió összefésülő* (*region merging*) módszerekben előfeldolgozó lépésként is szokták alkalmazni. A lokális minimum-helyek (a medence-fenek) száma csökkenthető, ha a kiindulási képen (Gauss-)simítást alkalmazunk. Az 5.5. ábra kétféle mértékű simításra mutatja be a vízválasztó szegmentálást.



5.5. ábra. Példa vízválasztó szegmentálással kapott régiók számának csökkentésére. A kiindulási képre (bal) a  $\sigma_1$  paraméterű Gauss simítást követően kapott régiók (középső) és a  $\sigma_2$  paraméterrel kapottak (jobb), ahol  $\sigma_2 > \sigma_1$ .

A vízválasztó szegmentálást gyakran alkalmazzák összeérő (vagy egymást részben átfedő) objektumok szétvágására. Ekkor háttérpontokból (mint jellemző pontokból) számított távolságtérkép negatívjából indulunk ki, vagyis a távolságtérkép csúcspontjaiból növesztjük a régiókat (ld. 5.6. ábra)



5.6. ábra. Példa összeérő bináris objektumok szétvágására vízválasztó szegmentálással. A kiindulási bináris kép (bal), a távolságtérkép (középső) és a szegmentálás eredménye (jobb), ahol a szétvágások gátszakaszainak a fehér vonalak felelnek meg.

A vízválasztó szegmentálás főbb megközelítéseinek (jelesül a Vincent-Soille, a legrövidebb út, a hegymászó és a soros bejárás algoritmus) részletes ismertetése szétfeszítené a jegyzet kereteit. A terület iránt mélyebben érdeklődők figyelmébe Roerdink és Meijster áttekintő művét [51] és a

<http://cmm.enscm.fr/~beucher/wtshed.html> oldalt ajánljuk.

### 5.3. Szegmentálás chamfer illesztéssel

Az illesztési módszerekkel ismert tárgyak, minták előfordulásait keressük a képen egy alkalmas illesztési kritérium (pl. kereszt-korreláció) optimalizálásával. A távolságtérképből származtatott illesztési kritériumot Borgefors javasolta [11]. A módszer alkalmazásánál először a bináris éltérképre chamfer távolságtérképet számítunk, ahol a jellemző pontok az élpontok. (A 3. fejezetben bemutatott Borgefors algoritmusát [10], mely (a képpontok száma szerinti) lineáris időben számít chamfer-távolságtérképet tetszőleges dimenzióban.) Ezt követően a keresett minta/szegmens kontúrpointjait helyezzük rá eltolva minden képpozícióra. A keresett minta előfordulását ott detektáljuk, ahol a távolságtérképnek a minta kontúrpointjaival talál-

0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	0	1	0
0	0	0	0	1	0	1	0
0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

5	4	3	2	1	1	1	2
4	3	2	1	0	0	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	0	0	1
5	4	3	2	1	1	1	2
6	5	4	3	2	2	2	3
7	6	5	4	3	3	3	4

5.7. ábra. Példa chamfer illesztésre. A kiindulási bináris éltérkép (bal) és az élpontokra mint jellemző pontokra számított távolságtérkép (jobb) a 4-távolság (mint speciális chamfer távolság) alapján.

5	4	3	2	1	1	1	2
4	3	2	1	0	0	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	0	0	1
5	4	3	2	1	1	1	2
6	5	4	3	2	2	2	3
7	6	5	4	3	3	3	4

5	4	3	2	1	1	1	2
4	3	2	1	0	0	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	0	0	1
5	4	3	2	1	1	1	2
6	5	4	3	2	2	2	3
7	6	5	4	3	3	3	4

5.8. ábra. Egy  $4 \times 3$ -as téglalap mint minta előfordulásainak keresése az 5.8. ábra távolságtérképén. A bal alsó sarokban (bal) a minta kontúrponthaival fedésbe kerülő távolságértékek összege 45, míg a minta egy másik helyre történő eltolásával (jobb) az illesztési kritérium 13.

kozó pozícióin egy alkalmasan megválasztott küszöbérték alatt marad az elemek (távolságok) összege.

A módszert az 5.7., az 5.8. és az 5.9. ábrákkal szemléltetjük.

5	4	3	2	1	1	1	2
4	3	2	1	0	0	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	0	0	1
5	4	3	2	1	1	1	2
6	5	4	3	2	2	2	3
7	6	5	4	3	3	3	4

5	4	3	2	1	1	1	2
4	3	2	1	0	0	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	1	0	1
4	3	2	1	0	0	0	1
5	4	3	2	1	1	1	2
6	5	4	3	2	2	2	3
7	6	5	4	3	3	3	4

5.9. ábra. Egy  $4 \times 3$ -as téglalap mint minta előfordulásainak keresése az 5.8. ábra távolságtérképén. A bal oldali mintapozíciónál az illesztési kritérium értéke 6, míg a jobb oldalinál 0, vagyis egy pontos illesztést találtunk.

A módszer megbízhatóan működik akkor is, ha az éltérképen hiányos vagy nem pontos a keresett tárgy kontúrja, viszont (csakúgy, mint bármely más mintaillesztő eljárás) csak a minta deformálatlan példányaait találja meg. Amennyiben pl. a minta más méretű vagy irányú előfordulásait keressük, úgy az eljárást meg kell ismételni a transzformált mintákra.

Megjegyezzük, hogy piramis struktúrával (ld. 10. fejezet) csökkenthető a műveleti komplexitás: Először egy durvább (lekicsinyített) mintát illesztünk egy durvább piramis-szinten, majd már csak azon pozíciókat (és szűk környezetüket) vizsgáljuk finomabb felbontásban, amelyekre a minimalizálandó illesztési kritérium alatta maradt egy bizonyos küszöbnek.

## 5.4. Szegmentálás fuzzy megközelítéssel

A fuzzy (rész)halmazok elméletét Zadeh vezette be a múlt század hatvanas éveiben [80]. Legyen  $X$  egy referencia halmaz, mely nem az  $\mathcal{A}$  fuzzy részhalmaza:

$$\mathcal{A} = \{ (x, \mu_{\mathcal{A}}(x)) \mid x \in X \},$$

ahol

$$\mu_{\mathcal{A}} : X \rightarrow [0,1]$$

az  $\mathcal{A}$  tagsági függvénye (*membership function*).

A szokványos halmazoknál a tagsági függvény helyett a karakterisztikus függvény elnevezést szoktuk alkalmazni, mely a referencia halmaz valamely elemére 1, ha az illető elem tartalmazza a kérdéses részhalmaz, különben a karakterisztikus függvény értéke 0. A fuzzy részhalmazok tagsági függvényeinek értelmezési tartománya a diszkrét 0 és 1 értékekkel szemben a  $[0,1]$  intervallum.

A fuzzy részhalmazokra a szokványos fogalmak és halmazműveletek:

- $\mathcal{A} \neq \emptyset \iff \exists x \in X : \mu_{\mathcal{A}}(x) \neq 0$
- $\overline{\mathcal{A}} = \{ (x, \mu_{\overline{\mathcal{A}}}(x)) \mid x \in X \},$   
ahol  $\forall x \in X : \mu_{\overline{\mathcal{A}}}(x) = 1 - \mu_{\mathcal{A}}(x)$
- $\mathcal{A} \cup \mathcal{B} = \{ (x, \mu_{\mathcal{A} \cup \mathcal{B}}(x)) \mid x \in X \},$   
ahol  $\forall x \in X : \mu_{\mathcal{A} \cup \mathcal{B}}(x) = \max\{\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)\}$
- $\mathcal{A} \cap \mathcal{B} = \{ (x, \mu_{\mathcal{A} \cap \mathcal{B}}(x)) \mid x \in X \},$   
ahol  $\forall x \in X : \mu_{\mathcal{A} \cap \mathcal{B}}(x) = \min\{\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)\}$

Fuzzy relációkat szintén definiálhatunk, mivel a relációk is (rész)halmazok. Egy  $p$  fuzzy reláció  $X^2$  fölött a következőképpen definiált:

$$p = \{ ((x, y), \mu_p(x, y)) \mid (x, y) \in X^2 \},$$

ahol

$$\mu_p : X^2 \rightarrow [0,1].$$

A fuzzy relációk között is megkülönböztethetünk speciálisakat:

- **reflexív:**  $\forall (x, x) \in X^2, \mu_p(x, x) = 1,$
- **szimmetrikus:**  $\forall (x, y) \in X^2, \mu_p(x, y) = \mu_p(y, x),$
- **transzitiv:**  $\forall (x, y), (y, z), (x, z) \in X^2,$   
 $\mu_p(x, z) = \max_y \{\min\{\mu_p(x, y), \mu_p(y, z)\}\}.$

Digitális képekre Udupa és Samarasekera [71] terjesztette ki a fuzzy-elméletet. A  $(\mathbb{Z}^n, \alpha)$  rendezett pár egy fuzzy digitális tér (*fuzzy digital space*), ahol  $\mathbb{Z}^n$  a térelemek (*spatial element*) halmaza és

$$\mu_{\alpha} : \mathbb{Z}^n \times \mathbb{Z}^n \rightarrow [0,1]$$

egy reflexív és szimmetrikus *fuzzy szomszédsági* (*fuzzy adjacency*) reláció, melyre megkívánjuk a

$$\forall a, b, c \in \mathbb{Z}^n, \quad \|a - b\| < \|a - c\| \Rightarrow \mu_\alpha(a, b) \geq \mu_\alpha(a, c)$$

tulajdonság teljesülését, vagyis azt, hogy távolság növekedésével egyre csökkenjen a térelemek közötti szomszédsági kapcsolat erőssége.

A  $(\mathbb{Z}^n, \alpha)$  fuzzy digitális tér felett a  $\mathcal{C} = (C, f)$  egy *fuzzy digitális kép*, ahol  $C \subseteq \mathbb{Z}^n$  a képpontok halmaza és az

$$f : C \rightarrow [0, 1]$$

függvény adja meg azok (normalizált) intenzitásait.

A  $(C, f)$  fuzzy digitális kép pontjai között a kapcsolatot a  $\kappa$  *fuzzy affinitás* (*fuzzy spel affinity*) reláció adja meg, ahol

$$\mu_\kappa(a, b) = \mathcal{F}(\mu_\alpha(a, b), f(a), f(b), a, b),$$

vagyis a kapcsolat erőssége függ(het) a fuzzy szomszédsági relációtól, az intenzitásoktól és a helytől is.

Fuzzy affinitásra példa:

$$\mu_\kappa(a, b) = \mu_\alpha(a, b) \cdot (1 - |f(a) - f(b)|).$$

A  $(\mathbb{Z}^n, \alpha)$  fuzzy tér feletti,  $\kappa$  fuzzy affinitású  $(C, f)$  fuzzy digitális kép pontjainak *fuzzy  $\kappa$ -összefüggőssége* (*fuzzy  $\kappa$ -connectedness*) egy újabb fuzzy reláció:

$$\mu_K(a, b) = \max_{\substack{\langle c_0 c_1 \dots c_m \rangle \\ c_0 = a, c_m = b}} \left\{ \min \left\{ \mu_\kappa(c_0, c_1), \dots, \mu_\kappa(c_{m-1}, c_m) \right\} \right\},$$

vagyis az összefüggősséget a pontokat összekötő legerősebb út/lánc ereje adja meg, ahol egy lánc ereje megegyezik a leggyengébb láncszemének erejével (a láncszem két pontja közötti fuzzy affinitással).

Egy  $(C, f)$  fuzzy digitális kép *fuzzy  $\kappa$ -komponense* a  $K_\Theta$  ( $\Theta \subseteq [0, 1]$ ) nem-fuzzy relációval adott:

$$\mu_{K_\Theta}(a, b) = \begin{cases} 1 & , \text{ ha } \mu_K(a, b) \in \Theta \\ 0 & , \text{ különben} \end{cases}.$$

Könnyen belátható, hogy a fenti reláció ekvivalencia-reláció, vagyis a  $C$  halmaz egy osztályozását adja. Az  $x \in C$  elemet tartalmazó ekvivalencia-osztályt  $[x]_{K_\Theta}$ -vel jelöljük. Az ekvivalencia-osztályok a kép *fuzzy objektumai*, melyekre:

$$[x]_{K_\Theta} = \left\{ y \mid \mu_{K_\Theta}(x, y) = 1 \right\}.$$

A  $(C, f)$  fuzzy digitális kép minden egyes  $x \in C$  pontjára képezhetünk egy képet, az  $f_x$  *fuzzy  $\kappa$ -összefüggősségi térkép* (*fuzzy  $\kappa$ -connectedness map*):

$$f_x(y) = \mu_K(x, y) \quad \text{valamennyi } y \in C \text{ pontra.}$$

Udupa és Samarasekera [71] a fuzzy összefüggősségi térkép meghatározására az alábbi (dinamikus programozási technikát követő) algoritmust javasolta:



**Algoritmus:** fuzzy összefüggőségi térkép számítása

*Input:* a  $(C, f)$  fuzzy digitális kép,

az  $x \in C$  pont és

az a  $\kappa$  fuzzy affinitás

*Output:* a  $(C, f_x)$  fuzzy  $\kappa$ -összefüggőségi térkép

tegyük be az  $x$  pontot az üres  $Q$  sorba

legyen  $f_x(x) = 1$  és legyen  $f_x(y) = 0$  minden  $y \in C, y \neq x$  pontra

while  $Q$  nem üres do

vegyük ki az  $y$  elemet  $Q$ -ból

$m = \max_z \{ \min \{ f_x(z), \mu_\kappa(d, z) \} \}$

if  $m > f_x(y)$  then

$f_x(y) = m$

tegyük be  $Q$ -ba mindazon  $z \in C$  pontokat, melyre  $\mu_\kappa(y, z) > 0$

A fuzzy-szegmentálásnál az  $x$  referenciapontot tartalmazó szegmenst a  $(C, f_x)$  fuzzy összefüggőségi térkép egyszerű küszöbölésével kapjuk. Herman [28] az alábbi  $\alpha$  és a  $\kappa$  relációkat alkalmazta:

$$\mu_\alpha(a, b) = \begin{cases} 1, & \text{ha } ||a - b|| \leq 1 \\ 0, & \text{különben} \end{cases}.$$

és

$$\mu_\kappa(a, b) = \begin{cases} \mu_\alpha(a, b) \cdot [w_1 h_1(f(a) + f(b)) + w_2 h_2(|f(a) - f(b)|)], & \text{ha } a \neq b \\ 0, & \text{ha } a = b \end{cases},$$

ahol

$$w_1 + w_2 = 1$$

és

$$h_i(x) = \exp \left( -\frac{(x - \mu_i)^2}{2\sigma_i^2} \right) \quad (i = 1, 2),$$

ahol  $\mu_1, \sigma_1$  a referenciapont körüli érdekes régió (ROI) intenzitásainak átlaga és szórása,  $\mu_2, \sigma_2$  pedig ugyanazon ROI átlaga és szórása a gradiens magnitúdó képen.

A fuzzy objektumok és a szegmentálás szemléltetésére nézzük a következő (mindössze  $2 \times 2$ -es) egyszerű példát:

$$C : \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad f : \begin{bmatrix} 0.1 & 0.2 \\ 0.5 & 0.8 \end{bmatrix}$$

Példánkon az  $\alpha$  fuzzy szomszédsági reláció legyen

$$\mu_\alpha(x, y) = \begin{cases} 1, & \text{ha } x = y \\ 0.5, & \text{ha } x \text{ és } y \text{ 4-szomszédosak} \\ 0, & \text{különben} \end{cases}.$$

és a  $\mu_\kappa$  fuzzy affinitás reláció legyen

$$\mu_\kappa(x, y) = \mu_\alpha(x, y) \cdot (1 - |f(x) - f(y)|).$$

Ekkor a példaképünkre a  $\alpha$  és a  $\kappa$  relációk táblázatai:

$\mu_\alpha$	a	b	c	d
a	1.0	0.5	0.5	0.0
b	0.5	1.0	0.0	0.5
c	0.5	0.0	1.0	0.5
d	0.0	0.5	0.5	1.0

$\mu_\kappa$	a	b	c	d
a	1.00	0.45	0.30	0.00
b	0.45	1.00	0.00	0.20
c	0.30	0.00	1.00	0.35
d	0.00	0.20	0.35	1.00

A  $\kappa$  fuzzy affinitás relációból a következő táblázatot kapjuk a  $K$  fuzzy összefüggés relációra:

$\mu_K$	a	b	c	d
a	1.00	0.45	0.30	0.30
b	0.45	1.00	0.30	0.30
c	0.30	0.30	1.00	0.35
d	0.30	0.30	0.35	1.00

A  $K$  fuzzy összefüggés reláció táblázatának soraiból kiolvashatjuk a fuzzy távolságtérképeket:

$$f_a : \begin{bmatrix} 1.00 & 0.45 \\ 0.30 & 0.30 \end{bmatrix} \quad f_b : \begin{bmatrix} 0.45 & 1.00 \\ 0.30 & 0.30 \end{bmatrix} \quad f_c : \begin{bmatrix} 0.30 & 0.30 \\ 1.00 & 0.35 \end{bmatrix} \quad f_d : \begin{bmatrix} 0.30 & 0.30 \\ 0.35 & 1.00 \end{bmatrix}.$$

A példánkon még adósak vagyunk a fuzzy  $\kappa$ -komponensekkel. Nézzük meg különböző  $\Theta$  halmazokra a  $K_\Theta$  nem-fuzzy ekvivalencia-relációk osztályozásait:

$K_\Theta$	a	b	c	d
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

$$\Theta = (0.45, 1] \\ \{a\}, \{b\}, \{c\}, \{d\}$$

$K_\Theta$	a	b	c	d
a	1	1	0	0
b	1	1	0	0
c	0	0	1	0
d	0	0	0	1

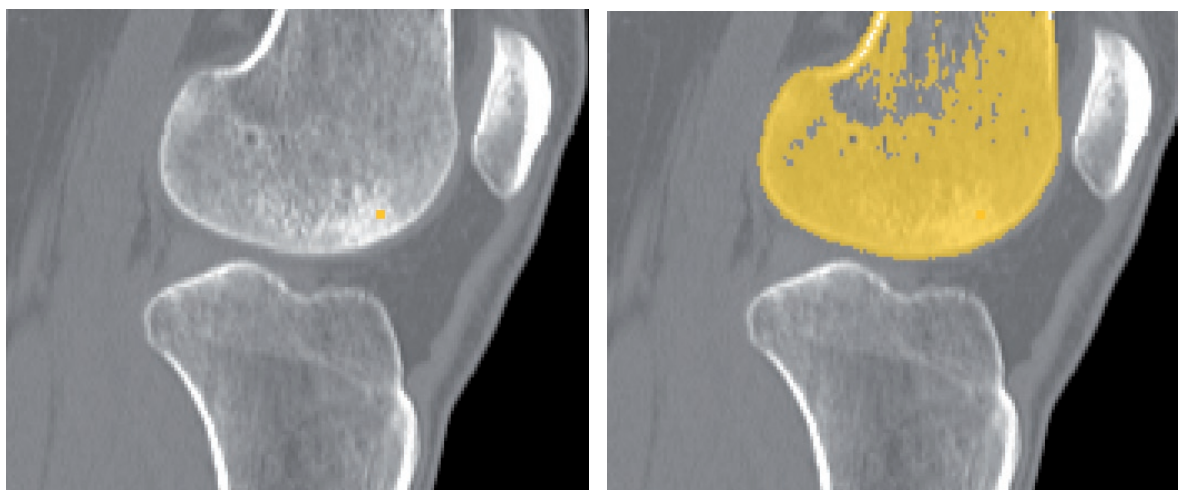
$$\Theta = (0.35, 1] \\ \{a, b\}, \{c\}, \{d\}$$

$K_\Theta$	a	b	c	d
a	1	1	0	0
b	1	1	0	0
c	0	0	1	1
d	0	0	1	1

$$\Theta = (0.30, 1] \\ \{a, b\}, \{c, d\}$$

$K_\Theta$	a	b	c	d
a	1	1	1	1
b	1	1	1	1
c	1	1	1	1
d	1	1	1	1

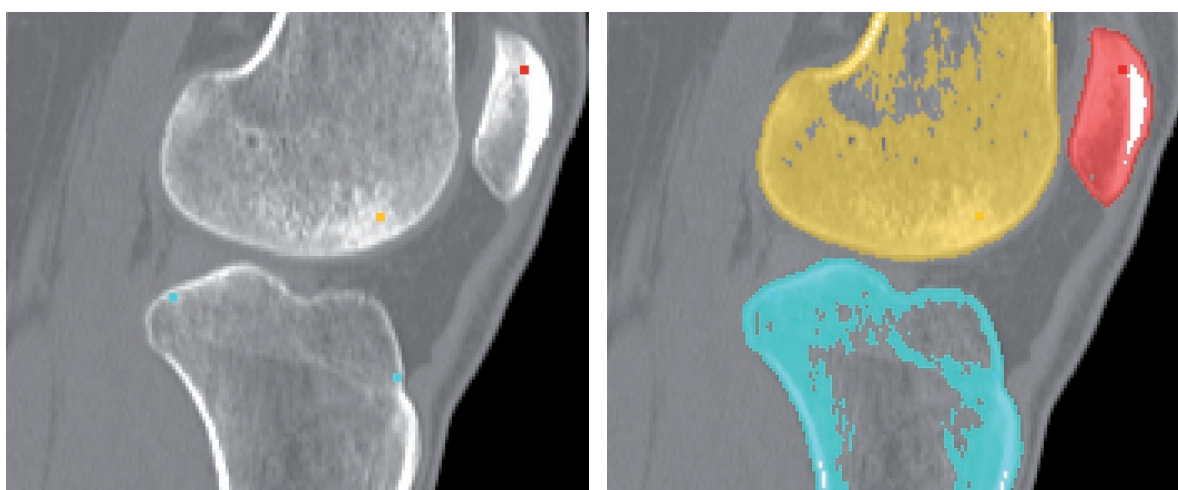
$$\Theta = [0, 1] \\ \{a, b, c, d\}$$



5.10. ábra. Példa fuzzy-szegmentálásra. Egy tédről készült CT felvétel és rajta a szegmentálandó objektum egy pontja (bal), a kijelölt forrást tartalmazó fuzzy objektum a kiindulási képre vetítve (jobb).

Az első esetben a négy képpont négy külön szegmensbe kerül. A második esetben a kép első sora már egy szegmens, a második sor két eleme pedig két egyelemű szegmenst alkot. A harmadikban a két sor két szegmenst formál, míg az utolsó esetben a teljes kép egyetlen szegmensből áll.

Megjegyezzük, hogy a fuzzy megközelítéssel történő szegmentálás intenzíven kutatott terület, kidolgoztak eljárásokat több forráspontra, egymással versengő régiókra is [17, 29]. Fuzzy-szegmentációra az 5.10. és az 5.11. ábrákkal adunk példát.



5.11. ábra. Példa fuzzy-szegmentálásra. Egy tédről készült CT felvétel és rajta három szegmentálandó objektum négy kijelölt pontja (bal), a kijelölt pontokat tartalmazó fuzzy objektumok a kiindulási képre vetítve (jobb).

A fuzzy technikáknak a digitális képfeldolgozásban betöltött szerepéről a <http://pami.uwaterloo.ca/tizhoosh/fip.htm> oldalon találhat még számos adalékot a terület iránt érdeklődő olvasó.

## 6. fejezet

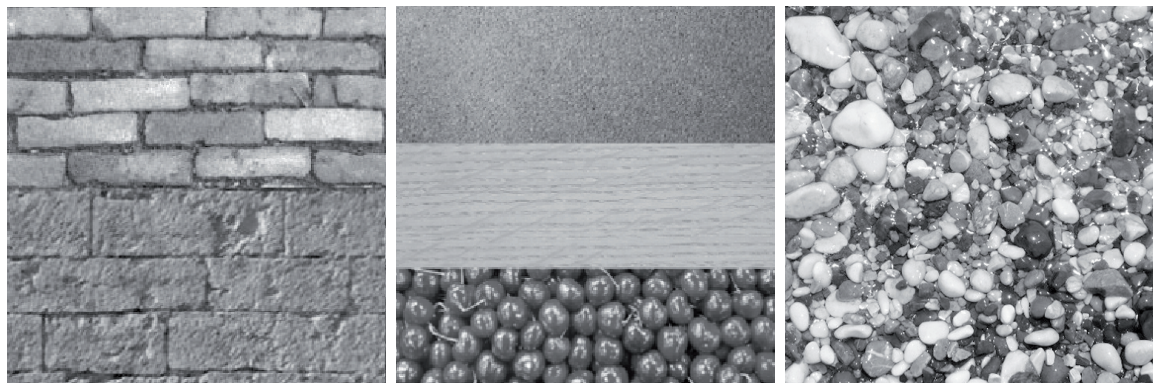
# Textúra-elemzés

A textúra kifejezést széles körben használják, éppen ezért nincs is egzakt meghatározása. A képfeldolgozásban textúrán az objektumok felszínére, szerkezetére, mintázatára vonatkozó tulajdonságokat értünk. A textúra egymással kölcsönöshatásban álló ismétlődő elemekből áll, melyeket a textúra primitíveknek vagy a textúra elemeknek (*texeleknek*) nevezünk.

A textúrákat jellemezhetjük (nem kvantitatív módon) szabályosnak (pl. egy kötés minta), véletlenszerűnek (pl. egy patak kavicsos medre), finomnak, durvának, szemcsésnek, vagy akár simának. A textúrákat leírhatjuk a texelek

- különböző mérete és eltérő fokú uniformitása,
- irányai,
- elhelyezkedései eltérő távolságokkal különböző irányokban,
- közötti eltérő kontraszt,
- eltérő háttere és
- eltérő fokú szabályossága alapján.

Textúrákra a 6.1. ábra mutat példákat.



6.1. ábra. Példák textúrákra.

A textúrák lehetnek gyengék és erősek. Gyenge textúránál csekély a primitívek közötti térbeli kölcsönhatás, míg erős textúránál a texelek valamiképpen szabályosan helyezkednek el.

A textúrák leírására javasolt módszereket két csoportba soroljuk: statisztikai és szintaktikus. A gyenge textúrák jellemzésére célszerű statisztikai módszereket választani, míg a szintaktikusak csak az erős textúrák esetén ígéretesek.

Lényeges megemlíteni azt, hogy a textúrák méretfüggők (pl. egy pepita zakó egy gyengébb felbontású képen középszürkének látszik), így a hierarchikus textúra-leírásra is szükség lehet, ahol egy adott skála-szint texeleinek valamely csoportját a következő (durvább felbontású) szinten texelként kezeljük.

A terület áttekintésénél Sonka, Hlavac és Boyle kézikönyvére [64], valamint Watt és Policarpo [77] művére támaszkodtunk. Az első alfejezetet a statisztikai módszereknek, a másodikat pedig a szintaktikusoknak szenteljük.

## 6.1. Statisztikai textúra-leírás

A következő két pontban a statisztikai textúra-elemzés két fő módszerét (a frekvenciákon és az együttes előfordulási mátrixokon alapulókat) tekintjük át. A szűkös terjedelem miatt nem térünk ki pl. az él-frekvenciákat, a Hurst-együtthatókat vagy a Laws-féle textúra-energia mértékeket használó megközelítésekre.

### 6.1.1. Frekvencián alapuló módszerek

A durvább textúrák nagyobb texeleket tartalmaznak, így alacsonyabb térbeli frekvenciákkal jellemezhetők. A finomabb textúrákra ennek a fordítottja teljesül (vagyis a kisebb textúrák vonzata a nagyobb térbeli frekvencia).

A térbeli frekvenciák jellemezhetők az  $f$   $M \times N$ -es kép *autokorrelációjával* (*autocorrelation*):

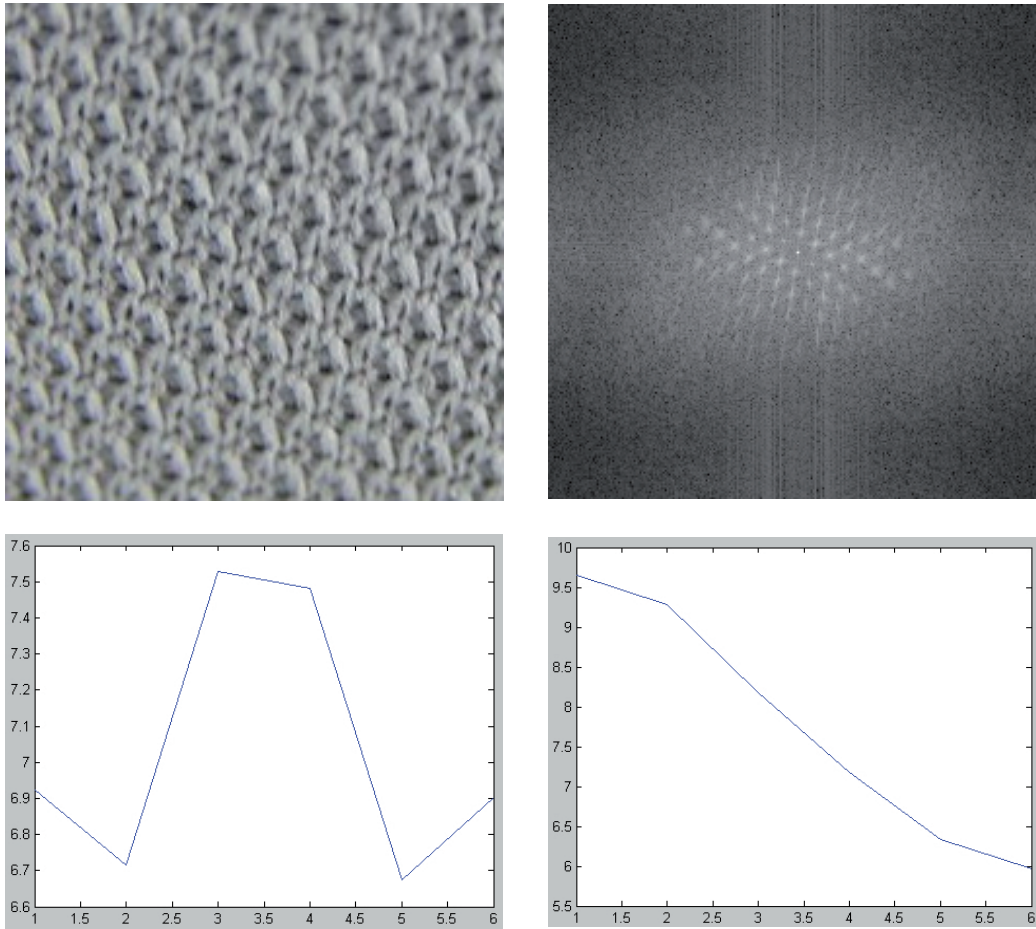
$$C_{ff}(p, q) = \frac{MN}{(M-p)(N-q)} \cdot \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} f(i, j) f(i+p, j+q)}{\sum_{i=1}^M \sum_{j=1}^N f(i, j)^2},$$

ahol a  $p$  és a  $q$  a képpont-pozíció eltérését adja meg a koordináta-tengelyek mentén. (Megjegyzendő, hogy az autokorreláció a frekvenciatérben is meghatározható a korrelációs tétel alapján.)

A  $C_{ff}$  mátrix elemei a korrelációs együtthatók. Durva textúrák esetén a korrelációs együtthatók értéke lassan csökken, a finomak esetén pedig gyorsan.

A térbeli frekvenciák természetesen jellemezhetők frekvencia térben is. A Fourier textúra analízis Fourier transzformált kép gyűrű- vagy ék-alakú tartományokban az  $|F(u, v)|$  Fourier-spektrumban mért átlagos energia-értékekkel jellemez. Az  $|F(u, v)|$  kép  $|F(r, \theta)|$





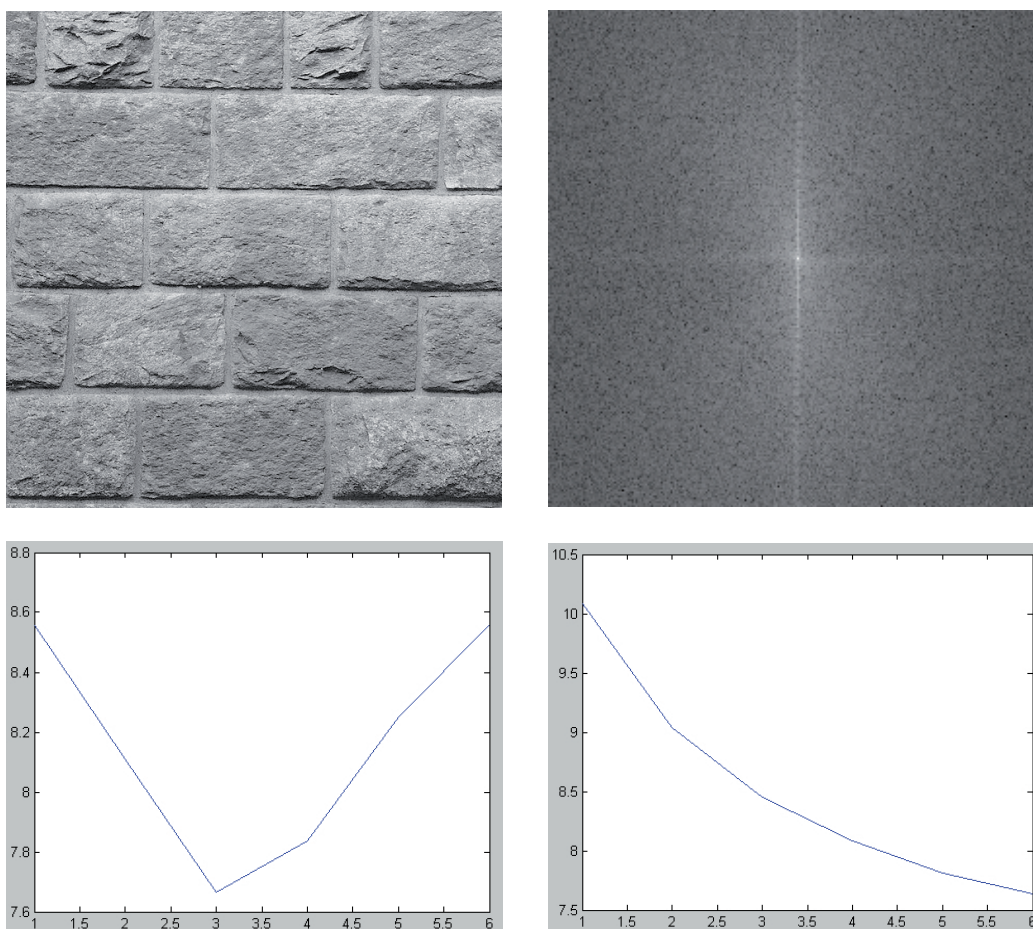
6.2. ábra. Példa Fourier textúra-jellemzésre. Egy textúra-kép (fent, bal), annak Fourier transzformáltja (fent, jobb), a frekvenciák összegzése bizonyos képátlók mentén (lent, bal) és a frekvenciák összegzése egy gyűrű alakú tartományban (lent, jobb). Formálisan: az alsó képek az  $S(\theta)$  függvény értékeit mutatják egy  $[\theta_1, \theta_2]$  szög-intervallumban (lent, bal) és az  $S(r)$  függvény értékei az  $[r_1, r_2]$  sugár-intervallumban (lent, jobb).

polárkoordinátás alakjából az alábbi két függvény származtatható:

$$S(r) = \int_{-\pi/2}^{\pi/2} |F(r, \theta)| d\theta,$$

$$S(\theta) = \int_0^R |F(r, \theta)| dr,$$

ahol  $R$  a képbe írható legnagyobb kör sugara. Az  $S(r)$  érték az alacsony és a magas frekvenciák eloszlását mondja meg (minden szögre) az  $r$ -sugarú (origó középpontú) körön, míg  $S(\theta)$  a  $\theta$  meredekségű egyenes mentén adja meg a frekvencia-eloszlást. Ezek az egyváltozós függvények értékes jellemzők a textúrák megkülönböztetések, osztályozásakor.



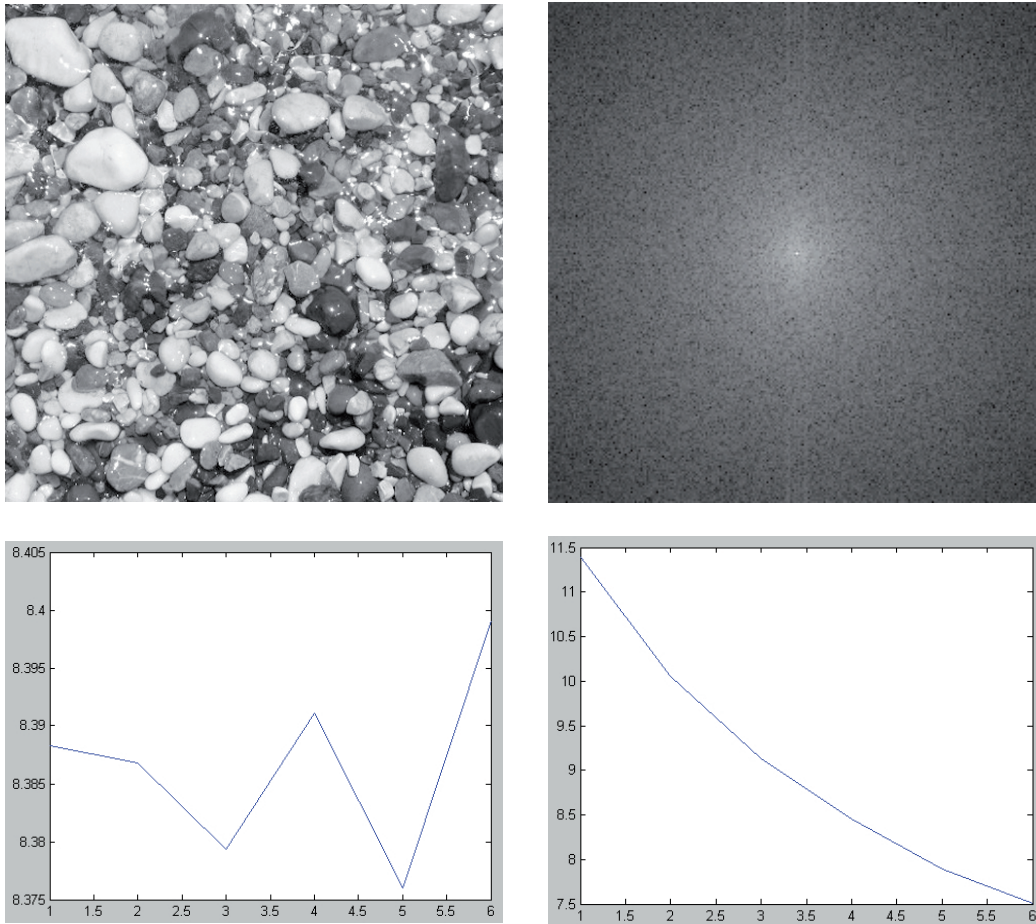
6.3. ábra. Példa Fourier textúra-jellemzésre. Egy textúra-kép (fent, bal), annak Fourier transzformáltja (fent, jobb), a frekvenciák összegzése bizonyos képátlók mentén (lent, bal) és a frekvenciák összegzése egy gyűrű alakú tartományban (lent, jobb). Ezúttal is ugyanazt a szög-intervallumot és sugár-intervallumot vettük, mint amelyekkel a 6.2. ábra készült.

Fourier textúra jellemzőkre a 6.2., a 6.3. és a 6.4. ábrákkal mutatunk példákat. Jól látható, hogy különböző textúra-képekre a Fourier textúra jellemzők markánsan eltérnek egymástól, így azok valóban alkalmazhatók a szegmentálás vagy a felismerés során.

### 6.1.2. Együttes előfordulási mátrixok

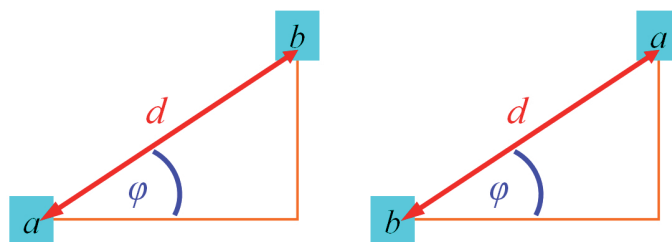
Az együttes előfordulási mátrixok (co-occurrence matrix) segítségével az intenzitási szintek változását másodrendű statisztikával írhatjuk le. Legyen  $L$  az  $f$  kép lehetséges intenzitásainak a halmaza és jelölje  $|L|$  az intenzitások, a kvantálási szintek számát (pl.  $L = \{0, 1, \dots, 255\}$ ,  $|L| = 256$ ). Az  $f$  képre a  $P_{\phi,d}$  együttes előfordulási mátrix  $P_{\phi,d}(a, b)$  eleme megadja azt, hogy a képen egymástól  $d$  távolságra és  $\phi$  szögben milyen gyakorisággal található  $a$  és  $b$  intenzitású képpont-pár (ld. 6.5. ábra).

A  $P_{\phi,d}$  együttes előfordulási mátrix szimmetrikus (vagyis  $P_{\phi,d} = P_{\phi,d}^T$ ) és a mérete  $|L| \times |L|$ . Együttes előfordulási mátrixokra a 6.6. és a 6.7. ábrák mutatnak példát.



6.4. ábra. Példa Fourier textúra-jellemzésre. Egy textúra-kép (fent, bal), annak Fourier transzformáltja (fent, jobb), a frekvenciák összegzése bizonyos képátlók mentén (lent, bal) és a frekvenciák összegzése egy gyűrű alakú tartományban (lent, jobb). Ezúttal is ugyanazt a szög-intervallumot és sugár-intervallumot vettük, mint amelyekkel a 6.2. és a 6.3. ábrák készültek.

Az együttes előfordulási mátrixok tehát információt hordoznak az intenzitás-variációkról a távolság és a szög függvényében. Minden egyes mátrixból számos globális jellemző származtatható, amelyeket Haralick-jellemzőknek is neveznek. Lássunk néhányat ezek közül:



6.5. ábra. Egymástól  $d$  távolságra és  $\phi$  szögben elhelyezkedő  $a$  és  $b$  intenzitású képpontok.

$$f = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix} \quad P_{0^\circ,1} = \begin{bmatrix} 2 & 2 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix}$$

6.6. ábra. Egy kép (bal) és annak együttes előfordulási mátrixa (jobb), ami az egymástól vízszintesen elhelyezkedő, egymással 4-szomszédos képpontok intenzitáspárjainak gyakoriságát tartalmazza. (Megjegyezzük, hogy az együttes előfordulási mátrixok mérete független a vizsgált képek méretétől.)

$$f = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix} \quad P_{135^\circ,1} = \begin{bmatrix} 1 & 1 & 3 & 0 \\ 1 & 1 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix}$$

6.7. ábra. Egy kép (bal) és annak együttes előfordulási mátrixa (jobb), ami a diagonális 8-szomszédok intenzitáspárjainak gyakoriságát tartalmazza.

Az *energia* (energy) (mint a szög szerinti második momentum) annál nagyobb értéket mutat, minél inkább homogén a kép:

$$\sum_{a,b} P_{\phi,d}^2(a,b).$$

Az *entrópia* (entropy):

$$\sum_{a,b} P_{\phi,d}(a,b) \cdot \log_2 P_{\phi,d}(a,b).$$

(Megjegyezzük, hogy az együttes előfordulási mátrixok entrópiájából származik az automatikus képregisztráció fontos hasonlósági mértéke is (ld. 8. fejezet).)

A *maximális valószínűség* (maximum probability) a leggyakoribb intenzitás-kombináció gyakorisága:

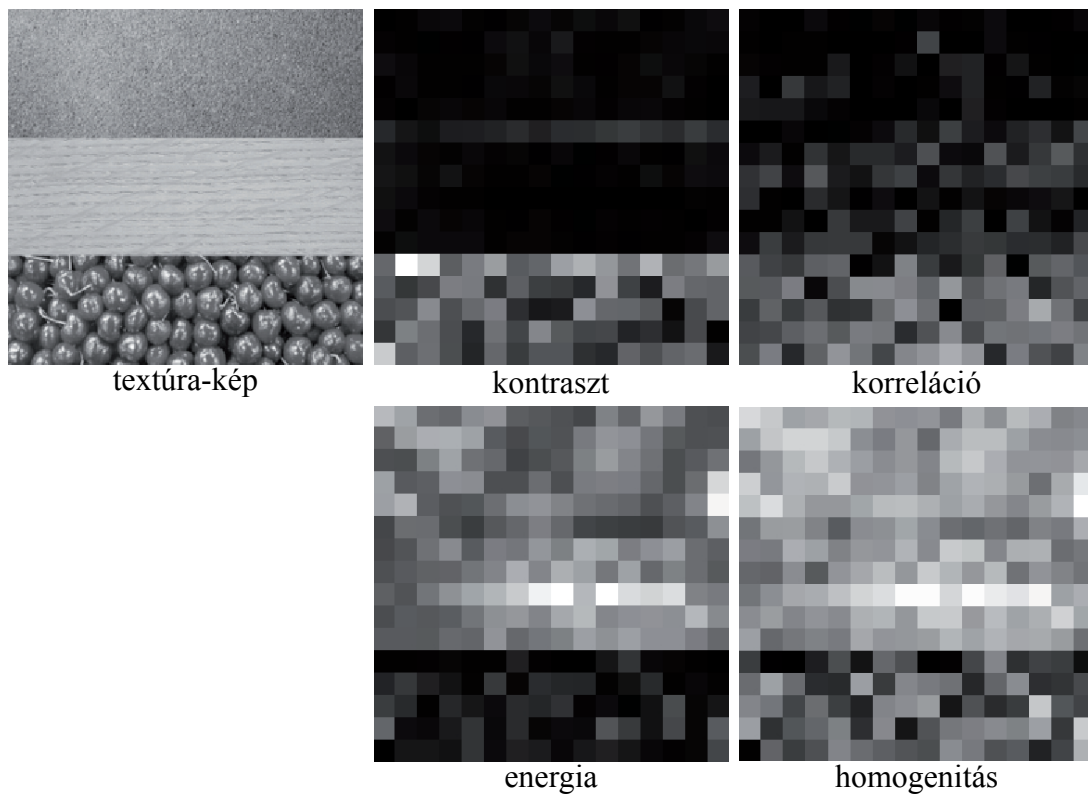
$$\max_{a,b} P_{\phi,d}(a,b).$$

A *kontraszt* (contrast) a lokális képvariációkat méri:

$$\sum_{a,b} |a-b|^\kappa \cdot P_{\phi,d}^\lambda(a,b).$$

A kontraszt két szabad paraméterének leggyakoribb megválasztása:  $\kappa = 2$ ,  $\lambda = 1$ .





6.8. ábra. Példa Haralick-jellemzőkre. Az együttes előfordulási mátrix paraméterei:  $d = 3$ ,  $\phi = 45^\circ$ .

Az inverz differencia momentum (*inverse difference moment*) – a kontraszthoz hasonlóan – ugyancsak két szabad paraméterrel bír:

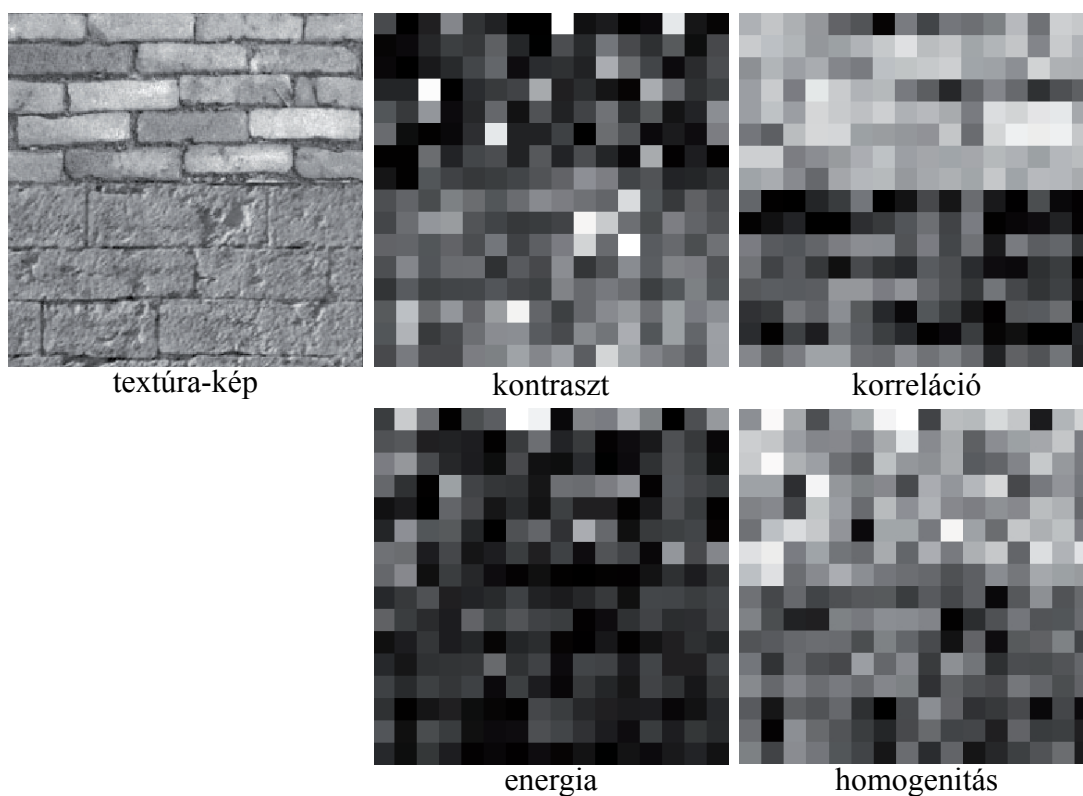
$$\sum_{\substack{a,b \\ a \neq b}} \frac{P_{\phi,d}^{\lambda}(a,b)}{|a-b|^{\kappa}}.$$

A homogenitás (*homogeneity*):

$$\sum_{a,b} \frac{P_{\phi,d}(a,b)}{1+|a-b|}.$$

A korreláció (*correlation*) a képi linearitást méri, a  $\phi$ -irányú struktúrák esetén ad nagyobb értékeket:

$$\frac{\sum_{a,b} [ab P_{\phi,d}(a,b)] - \mu_x \mu_y}{\sigma_x \sigma_y},$$



6.9. ábra. Példa Haralick-jellemzőkre. Az együttes előfordulási mátrix paraméterei:  $d = 2$ ,  $\phi = 180^\circ$ .

ahol

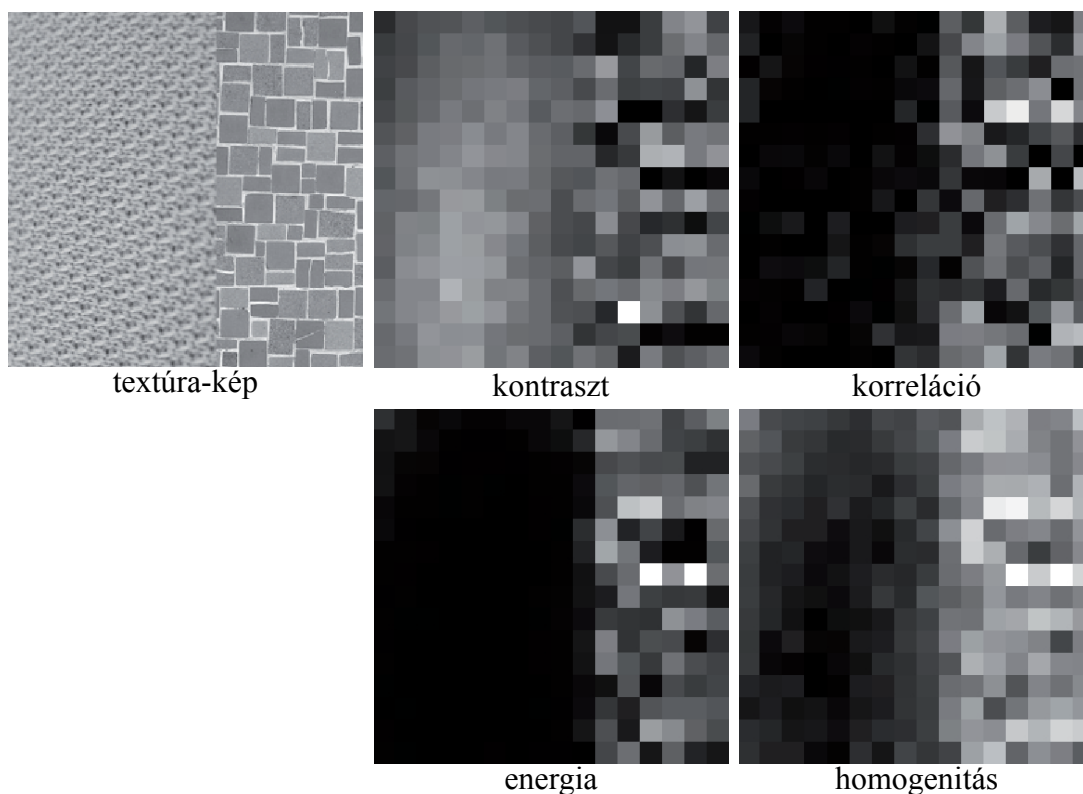
$$\begin{aligned}\mu_x &= \sum_a a \sum_b P_{\phi,d}(a, b), \\ \mu_y &= \sum_b b \sum_a P_{\phi,d}(a, b), \\ \sigma_x &= \sum_a (a - \mu_x)^2 \sum_b P_{\phi,d}(a, b), \\ \sigma_y &= \sum_b (b - \mu_y)^2 \sum_a P_{\phi,d}(a, b).\end{aligned}$$

Megjegyezzük, hogy a fenti jellemzőket általában nem a teljes képre, hanem a képpontok egy lokális környezetére (mint kisebb képekre) számítják, így a skalár jellemzőkből olyan képek/tömbök építhetők fel, amelyek alapján már ígéretesen szegmentálható a bemenő kép. Lokálisan számított Haralick-jellemzőkre a 6.8., a 6.9. és a 6.10. ábrákkal adunk példákat. Mindhárom ábrán a lokálisan számított jellemzőkből  $16 \times 16$ -os képeket állítottunk össze.

Az együttes előfordulási mátrixok kiszámítása valamennyi képpont-környezetre felettébb költséges. Gyorsabb, ha mátrixok helyett két vektort számítunk és belőlük származtatjuk a Haralick-jellemzők közelítését. A két vektor:

- $S_d$  az egymástól  $d$ -távolságra lévő képpontok összegének histogramja, és
- $D_d$  az egymástól  $d$ -távolságra lévő képpontok különbségének histogramja.





6.10. ábra. Példa Haralick-jellemzőkre. Az együttes előfordulási mátrix paraméterei:  $d = 2$ ,  $\phi = 0^\circ$ .

(Ha pl. a képünkre  $L = \{0, 1, \dots, 255\}$ , akkor az  $S_d$  és a  $D_d$  vektorok egyaránt 511-eleműek (mivel a két képpont összege 0 és 510, különbségük pedig  $-255$  és  $255$  közé eshet).

A négy legfontosabb Haralick-jellemzőt a fenti két vektorból a következő formulákkal közelíthetjük:

$$\begin{aligned}
 \text{kontraszt: } & \sum_j j^2 \cdot D(j), \\
 \text{homogenitás: } & \sum_j \frac{1}{1+j^2} \cdot D(j), \\
 \text{entrópia: } & -\sum_i S(i) \cdot \log(S(i)) - \sum_j D(j) \cdot \log(D(j)), \\
 \text{energia: } & \sum_j S(i)^2 \cdot \sum_j D(j)^2.
 \end{aligned}$$

A másodrendű statisztikán alapuló együttes előfordulási mátrixokkal végzett képjellemzés előnye, hogy térbeli tulajdonságokat is figyelembe vesz. A módszer korlátja, hogy a texelek alakját figyelmen kívül hagyja és nem alkalmazható nagyméretű primitívekre.

## 6.2. Szintaktikus textúra-leírás

A „szabályos” textúrák primitívjei között fennálló térbeli kapcsolat a formális nyelvek eszköztárával is leírható. A szintaktikus leírás az adott textúra-osztálynak megfeleltet egy generatív grammatikát. A környezetfüggetlen nyelvtanokban járatosak könnyen konstruálhatnak olyan nyelvtanokat, amelyek a szabályos mozaikok összefüggő részleteit ismerik fel.

A statisztikai módszerek általánosabbak, jobban elterjedtek, így a szintaktikus technikákat itt nem részletezzük.

## 7. fejezet

# Leképezések, warping, morphing

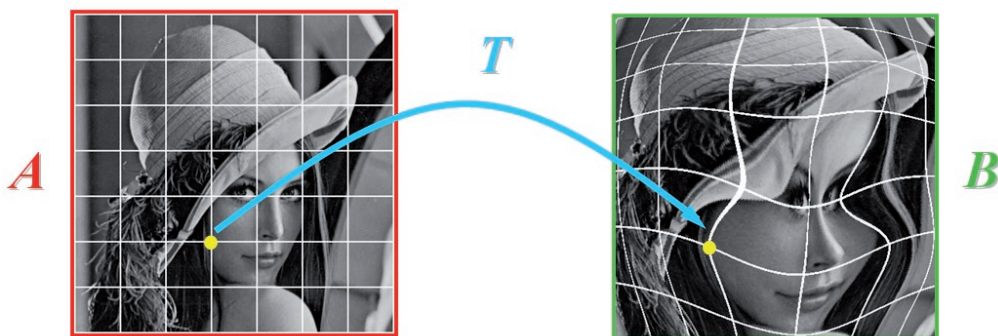
Pontokra a különféle geometriai transzformációk megadása és végrehajtása jól ismert [23], viszont a képek transzformációja (akár a skálázásuk is) közel sem egyszerű feladat, mivel egyrészt rácsponthoz képei (vagy inverz transzformációt nézve, az ősei) általában nem lesznek rácsponthoz, másrészt pedig a transzformált kép pontjainak intenzitását is meg kell határozni.

A jelen fejezet első pontjában a geometriai képtranszformációk, *leképezések* (*mapping*) végrehajtásával foglalkozunk. A második pontban a képek nem lineáris torzításának (*warping*) egy módszerét, a kijelölt kontrollpontokon alapuló TPS (*thin plate spline*) interpolációt mutatjuk be, majd a *képátalakítás* (*morphing*), vagyis a különböző képek egymásba átúsztatásának módszereit tekintjük át.

A fejezetben tárgyalt területek iránt érdeklődők figyelmébe Crane kézikönyvét [19] és Bookstein alapművét [9] ajánljuk.

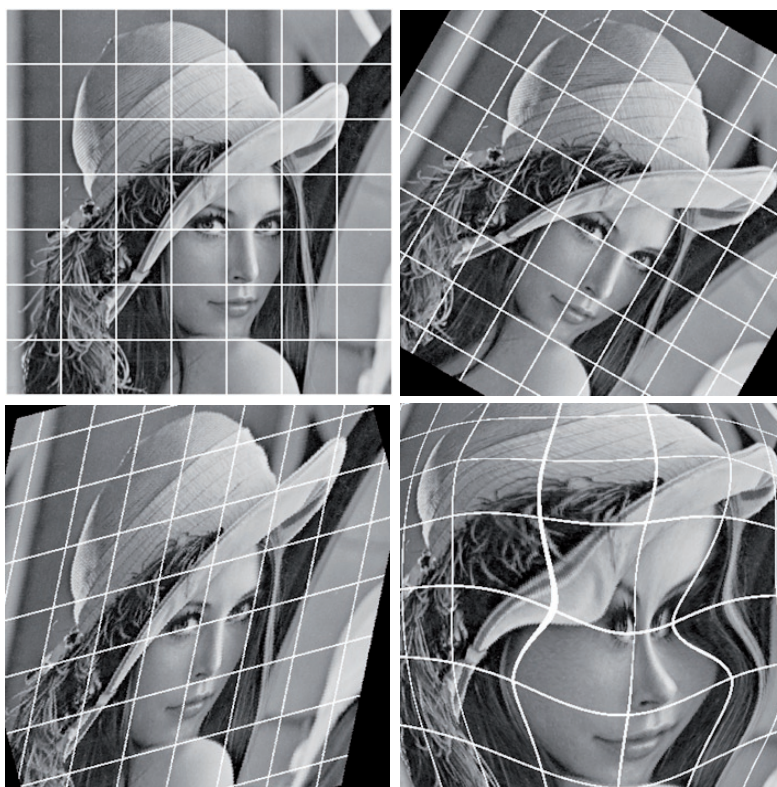
### 7.1. Képek geometriai transzformációi

Az  $A$  forrás képre a  $T$  geometriai transzformáció a  $B = T(A)$  cél képet eredményezi. A transzformáció a forrás kép újra-mintavételezésével jár (ld. 7.1. ábra).



7.1. ábra. Példa egy kép (nemlineáris) geometriai transzformációjára.

Képeknek – csakúgy, mint pontoknak – számos geometriai transzformációja ismert, melyek lehetnek parametrikusak vagy nem-parametrikusak, globálisak vagy lokálisak,... [23].

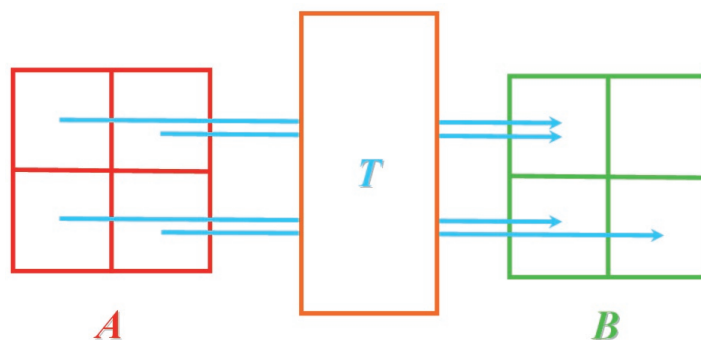


7.2. ábra. Példa 2-dimenziós geometriai transzformációkra. A forrás képből (fent, bal) egy merev-test (rigid body) (fent, jobb), egy affin (affine) (lent, bal) és egy nem-lineáris (nonlinear) (lent, jobb) transzformáltja. A 2-dimenziós merev test transzformációk esetén csak eltolás és elforgatás engedélyezett, így a transzformáció-típus szabadságfoka 3 (két eltolási paraméter a két tengely irányában és a vízszintes tengely elforgatási szöge az óramutató járásával ellentétes irányban). Az affin transzformációk családjába a párhuzamosságtartó lineáris transzformációk tartoznak, ahol a merev testhez (mint speciális esethez képest) engedélyezett még a skálázás és a nyírás is. A 2-dimenziós affin transzformációk szabadságfokát tekinthetjük 6-nak, mivel a homogén koordinátás transzformációs mátrixukban 6 változó/ismeretlen elem található [23]. A nem-lineáris transzformációk paraméterszáma akár több ezer is lehet.

A parametrikus transzformációkat osztályozhatjuk lineáris transzformációkra (melyeknél az egyenesek képe egyenes marad) vagy nem-lineárisra (ahol az egyenesek görbülnek). A parametrikus transzformációkat a szabadságfokuk is jellemzi, vagyis az, hogy hány paraméterük van. (A szabadságfok a paraméter-tér dimenziószáma. A paraméter-térben minden egyes pont egy konkrét transzformációnak felel meg.) A 7.2. ábrán 2-dimenziós transzformációkra láthatunk példákat.

A transzformáció végrehajtására alapvetően két megközelítés létezik: a *leképezés előre* (forward mapping) és a *leképezés hátra* (reverse mapping) (ld. 7.3. és a 7.5. ábra).

A leképezés előre módszer két hátrányos tulajdonsága: egyrészt a cél kép bizonyos területein a képpontoknak nincs ősük „üresen” maradhatnak (ld. 7.4. ábra), másrészt pedig olyan pontok is előfordulhatnak (pl. kicsinyítéskor), amelyeknél a forráskép ugyanazon pontja az ős.



7.3. ábra. A transzformáció végrehajtására leképezés előre módon.



7.4. ábra. Az  $(s_x, s_y) = (1, 2)$  skálázás leképezés előre megközelítéssel bal oldalon látható forrás képre. A cél kép (jobb) minden második oszlopa üresen maradt.

A leképezés hátra megközelítésnél viszont a cél képről lemaradhatnak a forrás kép bizonyos területei, továbbá a képpontoknak egynél több ősök is lehet. A gyakorlatban általában a leképezés hátra módszert alkalmazzák, melynek feltétele az, hogy létezzen  $T^{-1}$ , vagyis a kérdéses  $T$  transzformáció invertálható legyen. Az általános leképezés hátra eljárás a következőképpen működik:

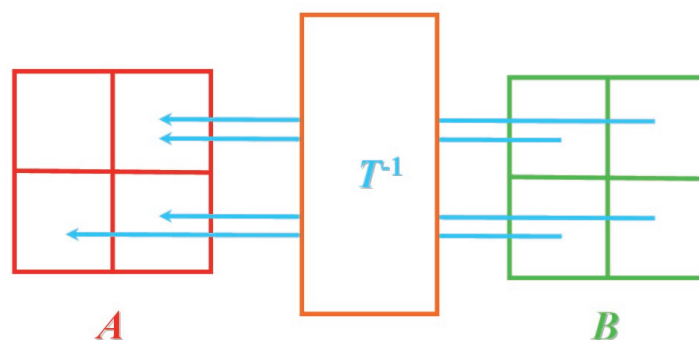
**Algoritmus:** leképezés hátra

*Input:* az  $A = [a(x, y)]$   $m \times n$ -es forrás kép

*Output:* a  $B = [b(x, y)]$   $m \times n$ -es cél kép

```
for  $x = 1$  to  $m$  do
  for  $y = 1$  to  $n$  do
    float  $u = T\_inv\_x(x, y)$ 
    float  $v = T\_inv\_y(x, y)$ 
     $b(x, y) = Interpolate(A, S, u, v)$ 
```

Az algoritmus pásztázza a cél képet és az megkeresi az aktuális  $(x, y)$  képpont ősének  $(u, v)$  koordinátáit az inverz transzformáció mellett. Hangsúlyozandó, hogy egy egész koordinátájú  $(x, y)$  pontra kapott  $(u, v)$  általában egy valós koordinátákkal rendelkező pont lesz, nem pedig a forrás kép valamely pontja. A célkép aktuális pontjának intenzitását így az  $(u, v)$  pont körüli  $S$  tartomány/környezetbe eső  $A$ -beli képpontok intenzitásai alapján határozzuk meg az *Interpolate* interpolációs függvénnyel.



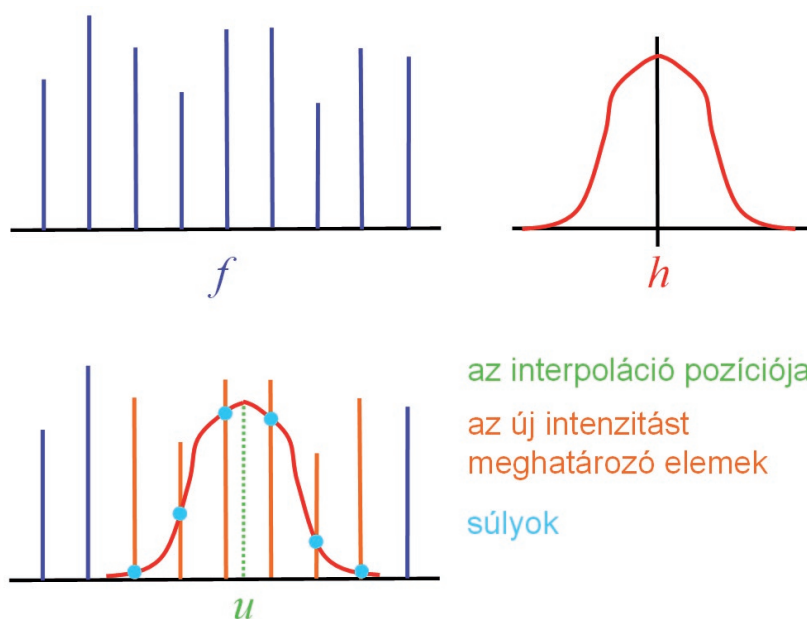
7.5. ábra. A transzformáció végrehajtására leképezés hátra megközelítéssel.

1-dimenziós esetben a  $g(x)$  valós függvény interpolációs függvény, teljesíti az alábbi feltételeket:

- $g(x) = g(-x)$  (vagyis  $g$  páros függvény, szimmetrikus a függőleges koordinátatengelyre),
- $\int_{-\infty}^{\infty} g(x) dx = 1$  (azaz a súlyok összege 1),
- $g(x) = 0$ , ha  $|x| > K$  (vagyis a  $g$  „lecseng” egy  $K$ -sugarú környezeten kívül).

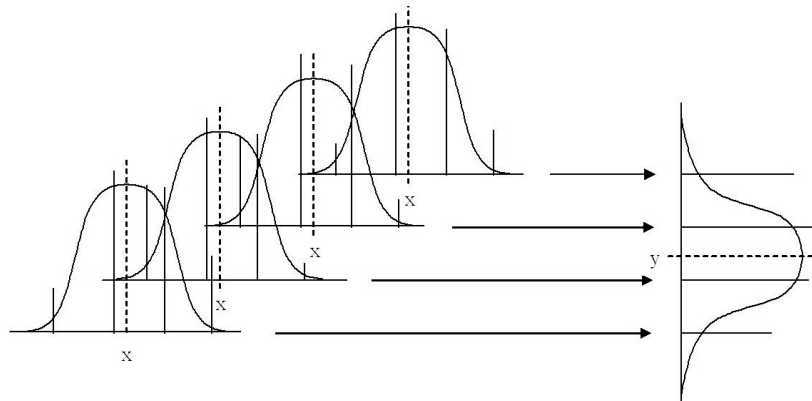
Az  $f$  analóg 1-dimenziós függvénynek a  $g$  interpolációs függvénnyel való interpoláltja:

$$h(x) = \int_{-\infty}^{\infty} f(u+x) \cdot g(x) dx,$$



7.6. ábra. Interpoláció 1-dimenziós diszkrét esetben.

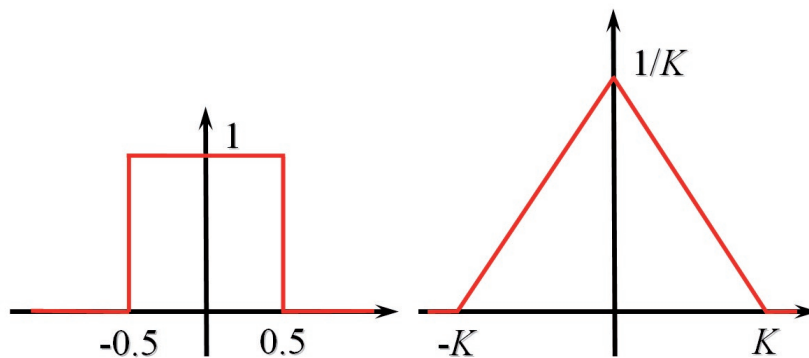




7.7. ábra. A 2-dimenziós interpoláció végrehajtható három 1-dimenziós interpolációval.

diszkrét  $f$  jelsorozatra pedig:

$$h(x) = \sum_{x=-K}^K f(\text{int}(u)+x) \cdot g(x - \text{frac}(u)).$$



7.8. ábra. A legközelebbi szomszéd (bal) és a lineáris (jobb) interpoláció függvényei.

Az 1-dimenziós interpoláció megértését a 7.6. ábra segíti.

A 2-dimenziós interpoláció helyettesíthető három 1-dimenziós interpolációval (ld. 7.7. ábra).

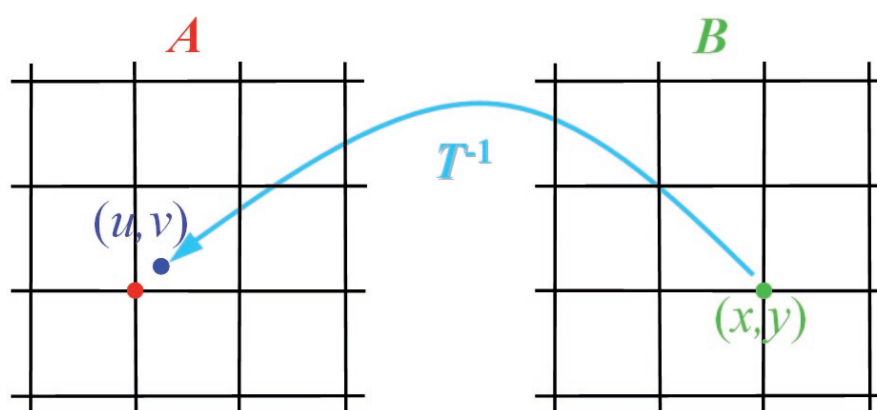
A legegyszerűbb interpolációs technikák a *legközelebbi szomszéd* (*nearest neighbor*) és a *(bi/tri)lineáris* (*(bi/tri)linear*) interpolációk, melyek függvényei a 7.8. ábrán láthatók.

A legközelebbi szomszéd interpoláció egyszerű kerekítést jelent, vagyis a leképezés hátra algoritmusának zárósora a következő:

$$b(x, y) = a(\text{round}(u), \text{round}(v)),$$

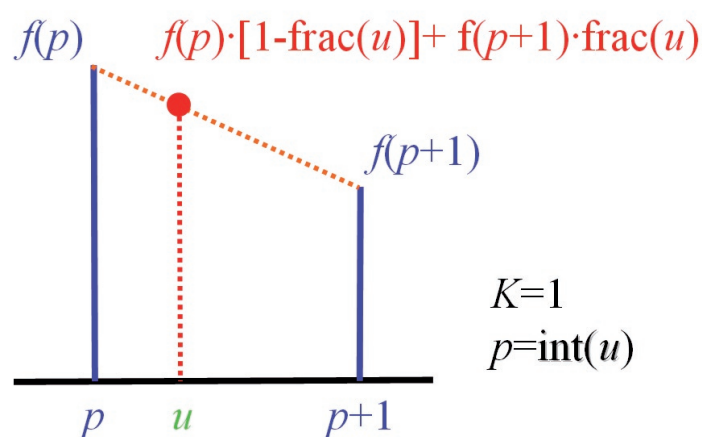
ahol *round* a legközelebbi egészre megadó kerekítő-függvény (ld. a 7.9. ábra).

Lineáris interpoláció esetén, az adott pozíció  $K$ -sugarú környezetébe eső képpontok a távolsággal lineárisan csökkenő súllyal járulnak hozzá a cél kép megfelelő pontjának inten-



7.9. ábra. Példa legközelebbi szomszéd interpolációra.

zítéséhez. A gyakorlatban leggyakrabban a  $K = 1$  értéket választjuk, vagyis a lineáris kombinációban 1-dimenziós esetben kettő, 2-dimenzióban pedig négy pont játszik szerepet (ld. a 7.10., a 7.11. és a 7.12. ábrákat).



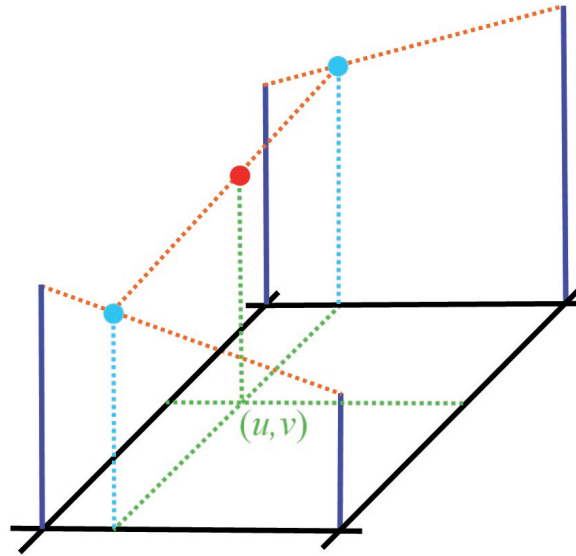
7.10. ábra. Lineáris interpoláció.

A legközelebbi szomszéd és a lineáris interpolációkénál bonyolultabb interpolációs függvényeket is javasoltak. Közülük itt kettőt, a *köbös* (*cubic*) és a *B-spline* interpolációkat ismeretjük.

Az 1-dimenziós köbös interpoláció függvénye:

$$g(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & 0 \leq |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases}$$

ahol az  $a$  paraméter javasolt értékei:  $-0.5$ ,  $-0.75$  és  $-1.0$ . A köbös interpolációs függvény a 7.13. ábrán látható.



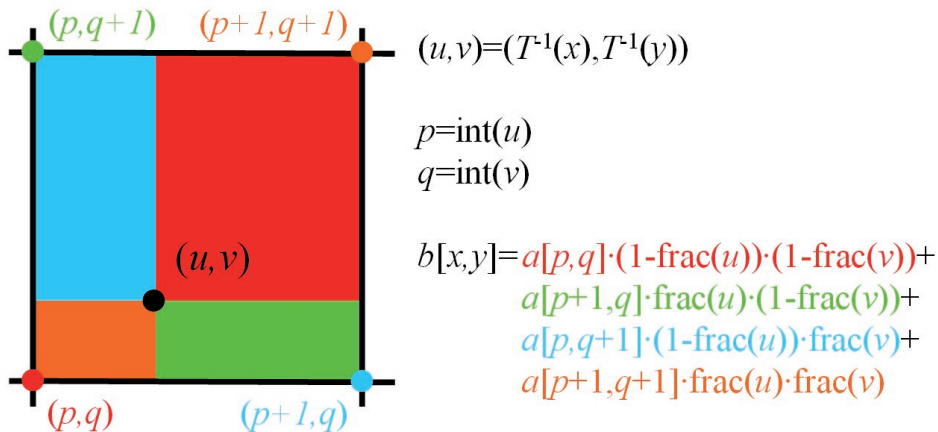
7.11. ábra. Bilineáris interpoláció (három 1-dimenziós lineáris interpolációval).

Az 1-dimenziós *B-spline* interpoláció függvény szintén harmadfokú:

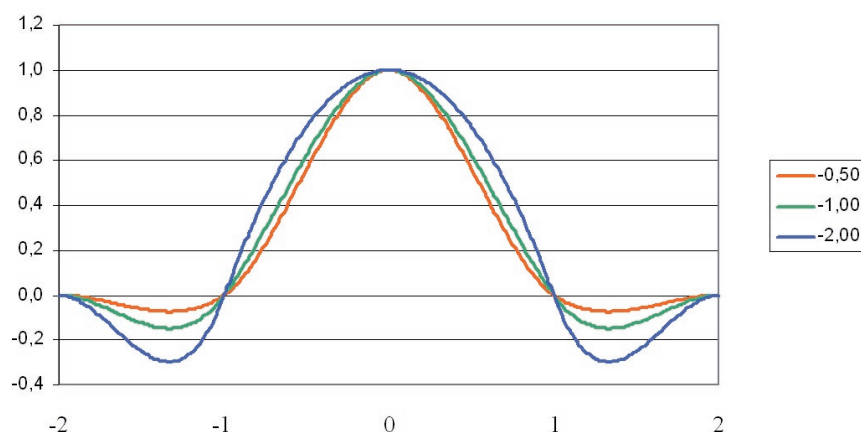
$$g(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3}, & 0 \leq |x| < 1 \\ -\frac{1}{6}|x|^3 + |x|^2 - 2|x| + \frac{4}{3}, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases}$$

Az interpoláció függvény a 7.14. ábrán látható.

A ismertetett négy interpolációs technikát a 7.15., a 7.16., a 7.17. és a 7.18. ábrákon hasonlítottuk össze. Ezúttal is a 7.2. ábrán (fent, bal) látható képet vettük forrás képnek. A teszt-



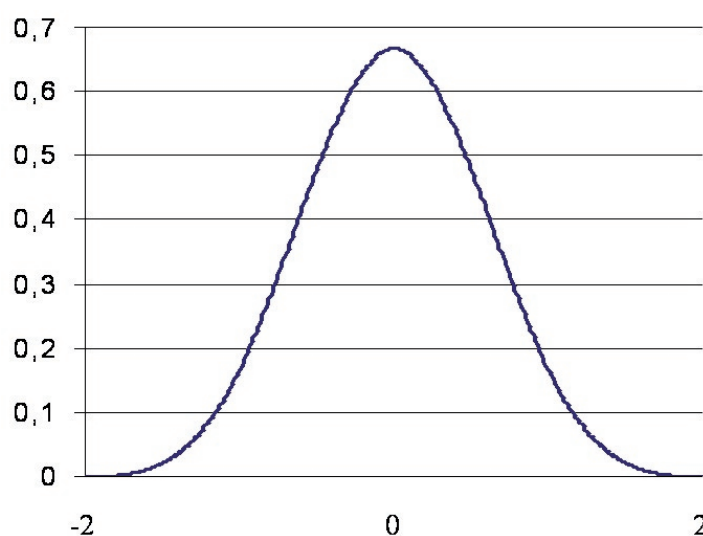
7.12. ábra. A bilineáris interpoláció számításának szemléltetése. A forrás képnek az egységnégyzet négy sarkában lévő intenzitásokat a pontokkal „szemközti” téglalap területével súlyozzuk. Ily módon a súlyok összege valóban 1. Megállapíthatjuk, hogy minél közelebb esik egy képponthoz az ős, annál nagyobb lesz a „szemközti” téglalap területe, vagyis a „közeli” képpontok intenzitásai „nagy” súllyal esnek latba.



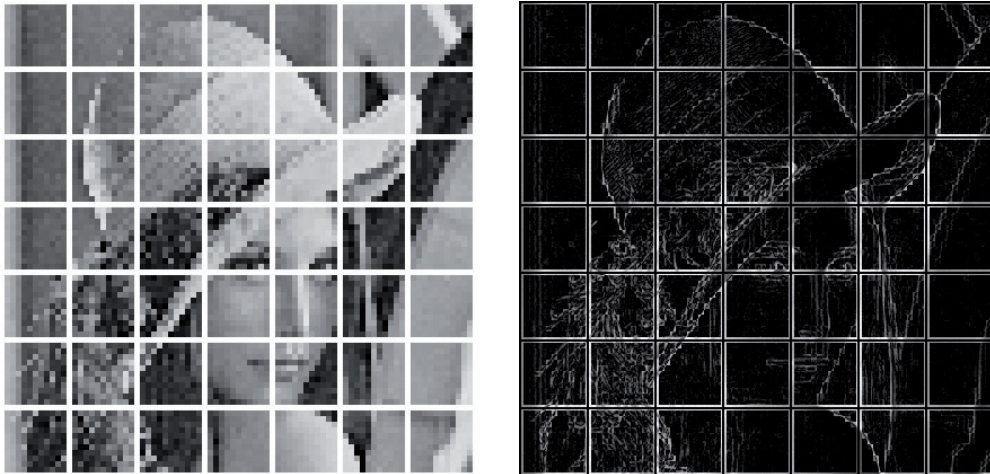
7.13. ábra. A köbös interpoláció függvényei ( $a = -0.5, -0.75, -1.0$ ). A függvény az origó körüli 2-sugarú tartományban nemnulla, így az interpolációval 4 pont intenzitásának súlyozott átlagát képezzük.

képet először az ötödére kicsinyítettük (vagyis az  $s_x, s_y = 0,2$  faktorokkal skáláztuk) a négy interpolációs technikával, majd a kapott képeket visszanyújtottuk az eredeti méretükre (az  $s_x, s_y = 5$  faktorokkal történő skálázással) ugyancsak a figyelembe vett interpolációval. Az összehasonlító ábrákon a kicsinyített majd felnagyított képek és a forrás képtől való eltéréseit mutatjuk be.

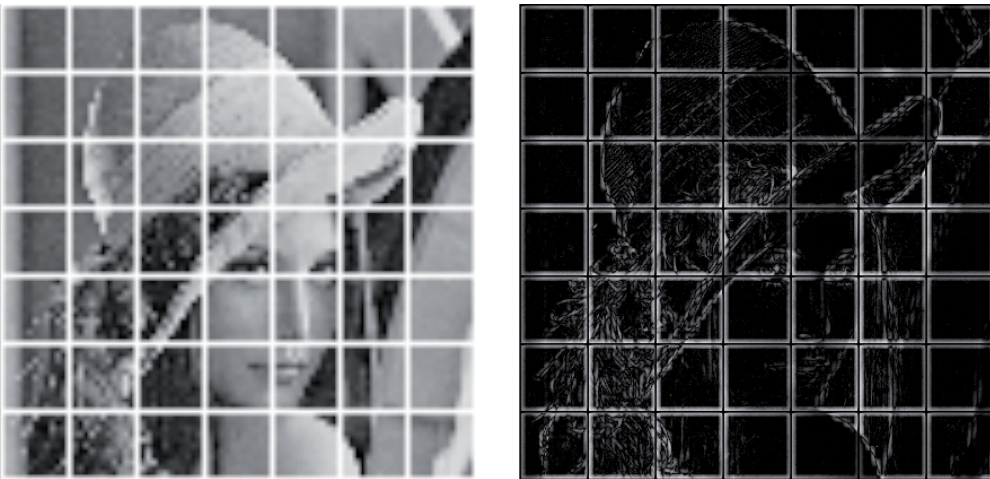
Megállapíthatjuk, hogy a legközelebbi szomszéd interpoláció a leggyorsabb és a legkönnyebben implementálható, további előnyös tulajdonsága, hogy megőrzi az éleket, viszont erősen blokkosít, darabossá teszi a képet. A többi tárgyalt interpolációnak simító, homályosító hatása van. A B-spline módszernél érhető tetten a legerőteljesebb homályosítás, így helyet-



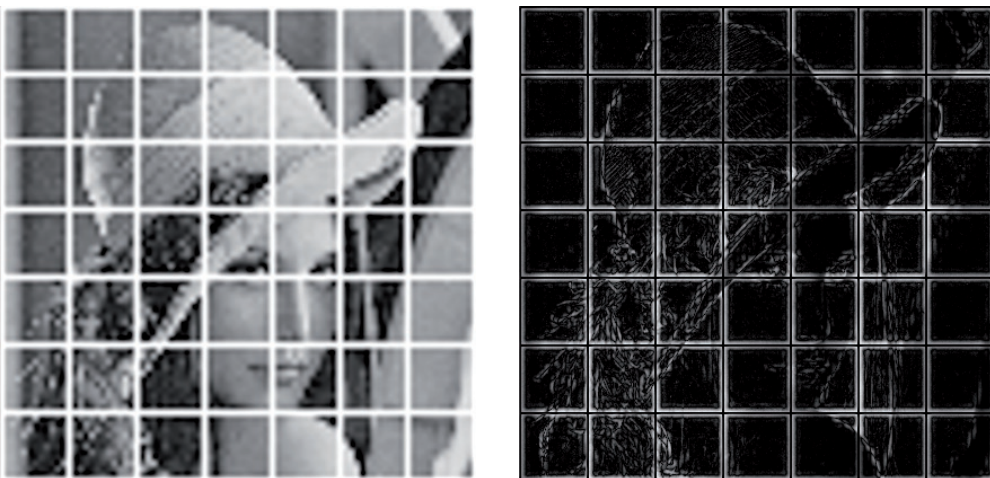
7.14. ábra. A B-spline interpoláció függvénye. Ennél a függvénynél is csak az origó körüli 2-sugarú nyílt intervallumban szerepelnek nemnulla, így az interpolációval ezúttal is 4 pont intenzitásának súlyozott átlagát képezzük.



7.15. ábra. A legközelebbi szomszéd interpoláció hatása.

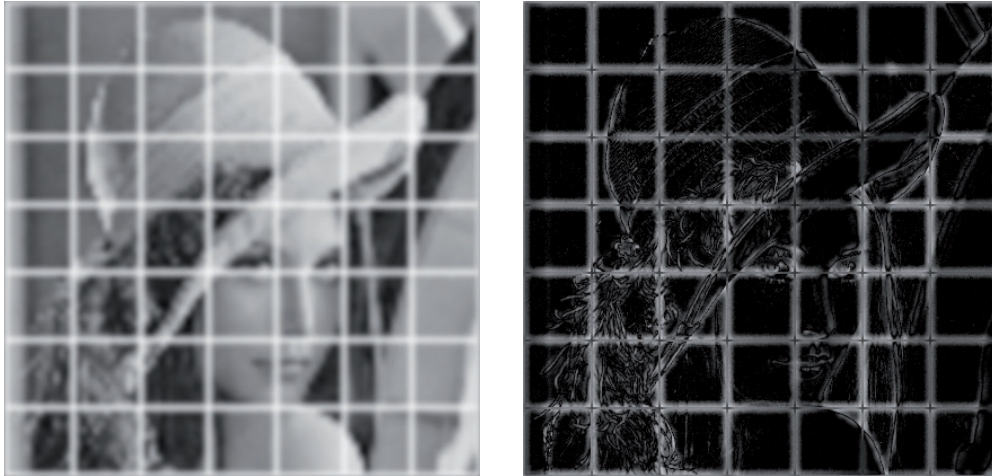


7.16. ábra. A bilineáris interpoláció hatása.



7.17. ábra. A köbös interpoláció hatása.





7.18. ábra. A B-spline interpoláció hatása.

te inkább a lineáris vagy a köbös javasolt. A képfeldolgozó, képmegjelenítő programokba leggyakrabban a lineáris interpolációt építik be.

## 7.2. Warping TPS interpolációval

A warping egy kép nemlineáris transzformációja vagy egy nemlineáris transzformáció animáciált végrehajtása (a forrás képnek a cél képbe történő deformációjáról egy mozgókép készül, ahol két szomszédos filmkocka között a változást egy „kisebb mértékű” nemlineáris transzformáció ad meg).

A következő alfejezetben az átalakításra adott módszerek a warping esetén is alkalmazhatók. Most a nemlineáris transzformációk egy kontrollpont-alapú megadásával, a Bookstein által javasolt TPS (*thin plate spline*) interpolációt ismertetjük 2-dimenziós és 3-dimenziós képekre [9].

A TPS interpoláció megadásához ki kell jelölnünk  $n$  pontpárt, az egymásnak megfeleltetett kontrollpontok  $\{P_1, P_2, \dots, P_n\}$  és  $\{Q_1, Q_2, \dots, Q_n\}$  halmazát, ahol 2-dimenziós esetben

$$P_i = (p_x^i, p_y^i), \quad Q_i = (q_x^i, q_y^i),$$

3-dimenzióban pedig

$$P_i = (p_x^i, p_y^i, p_z^i), \quad Q_i = (q_x^i, q_y^i, q_z^i)$$

( $i = 1, 2, \dots, n$ ).

A transzformációt 2-dimenzióban  $2(n+3)$ , 3-dimenzióban pedig  $3(n+4)$  (tehát a kontrollpontok számától függő számú) paraméter ad meg, melyekhez egy-egy lineáris egyenletrendszer megoldásával jutunk el. A keresett paraméterek 2-dimenziós esetben:

$$\begin{aligned} a, & \quad a_x, \quad a_y, \quad \alpha_1, \quad \alpha_2, \quad \dots, \quad \alpha_n, \\ b, & \quad b_x, \quad b_y, \quad \beta_1, \quad \beta_2, \quad \dots, \quad \beta_n, \end{aligned}$$



míg a 3-dimenziós esetben:

$$\begin{aligned} a, & a_x, a_y, a_z, \alpha_1, \alpha_2, \dots, \alpha_n, \\ b, & b_x, b_y, b_z, \beta_1, \beta_2, \dots, \beta_n, \\ c, & c_x, c_y, c_z, \gamma_1, \gamma_2, \dots, \gamma_n. \end{aligned}$$

A megoldandó egyenletrendszerek együttható-mátrixaiban szerepel a *biharmonikus egyenlet* (*biharmonic equation*) fundamentális megoldása, mely 2-dimenziós esetben

$$U(x, y) = (x^2 + y^2) \cdot \ln(x^2 + y^2),$$

3-dimenzióban pedig

$$U(x, y, z) = \sqrt{x^2 + y^2 + z^2},$$

vagyis érdekes módon az  $U$  függvény 3-dimenzióban egyszerűen az euklidészi norma, míg 2-dimenzióban egy bonyolultabb függvény.

A 2-dimenziós TPS egyenletrendszere:

$$\begin{bmatrix} U(Q_1 - Q_1) & U(Q_1 - Q_2) & \dots & U(Q_1 - Q_n) & 1 & q_x^1 & q_y^1 \\ U(Q_2 - Q_1) & U(Q_2 - Q_2) & \dots & U(Q_2 - Q_n) & 1 & q_x^2 & q_y^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ U(Q_n - Q_1) & U(Q_n - Q_2) & \dots & U(Q_n - Q_n) & 1 & q_x^n & q_y^n \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 \\ q_x^1 & q_x^2 & \dots & q_x^n & 0 & 0 & 0 \\ q_y^1 & q_y^2 & \dots & q_y^n & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \\ \vdots & \vdots \\ \alpha_n & \beta_n \\ a & b \\ a_x & b_x \\ a_y & b_y \end{bmatrix} = \begin{bmatrix} p_x^1 & p_y^1 \\ p_x^2 & p_y^2 \\ \vdots & \vdots \\ p_x^n & p_y^n \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

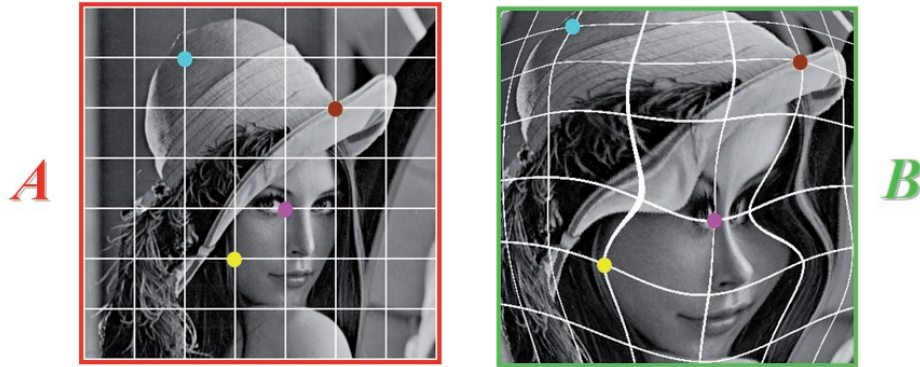
Az egyenletrendszer 3-dimenzióban:

$$\begin{bmatrix} U(Q_1 - Q_1) & U(Q_1 - Q_2) & \dots & U(Q_1 - Q_n) & 1 & q_x^1 & q_y^1 & q_z^1 \\ U(Q_2 - Q_1) & U(Q_2 - Q_2) & \dots & U(Q_2 - Q_n) & 1 & q_x^2 & q_y^2 & q_z^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ U(Q_n - Q_1) & U(Q_n - Q_2) & \dots & U(Q_n - Q_n) & 1 & q_x^n & q_y^n & q_z^n \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ q_x^1 & q_x^2 & \dots & q_x^n & 0 & 0 & 0 & 0 \\ q_y^1 & q_y^2 & \dots & q_y^n & 0 & 0 & 0 & 0 \\ q_z^1 & q_z^2 & \dots & q_z^n & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \vdots & \vdots & \vdots \\ \alpha_n & \beta_n & \gamma_n \\ a & b & c \\ a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{bmatrix} = \begin{bmatrix} p_x^1 & p_y^1 & p_z^1 \\ p_x^2 & p_y^2 & p_z^2 \\ \vdots & \vdots & \vdots \\ p_x^n & p_y^n & p_z^n \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Megállapíthatjuk, hogy az egyenletrendszerek bal oldala (tehát a megoldhatóság is) csak a  $Q_i$  pontoktól függ, míg a  $P_i$  kontrollpontok csak az egyenletrendszerek jobb oldalát befolyásolják.

Az egyenletrendszer megoldásával kapott paraméterekkel a koordinátánkénti transzformációs függvények 2-dimenzióban:

$$\begin{aligned} f_x(x, y) &= a + a_x \cdot x + a_y \cdot y + \sum_{i=1}^n \alpha_i \cdot U(Q_i - (x, y)), \\ f_y(x, y) &= b + b_x \cdot x + b_y \cdot y + \sum_{i=1}^n \beta_i \cdot U(Q_i - (x, y)), \end{aligned}$$



7.19. ábra. Példa TPS interpolációra, ahol a forrás képen (bal) a  $Q_i$  kontrollpontokat, a célképen (jobb) pedig a  $P_i$  kontrollpontokat jelöltük be ( $i = 1, 2, 3, 4$ ).

míg a 3-dimenziós TPS esetén:

$$f_x(x, y, z) = a + a_x \cdot x + a_y \cdot y + a_z \cdot z + \sum_{i=1}^n \alpha_i \cdot U(Q_i - (x, y, z)),$$

$$f_y(x, y, z) = b + b_x \cdot x + b_y \cdot y + b_z \cdot z + \sum_{i=1}^n \beta_i \cdot U(Q_i - (x, y, z)),$$

$$f_z(x, y, z) = c + c_x \cdot x + c_y \cdot y + c_z \cdot z + \sum_{i=1}^n \gamma_i \cdot U(Q_i - (x, y, z)).$$

A fenti egyenletek láttán három észrevétel kínálkozik:

- A transzformációk globálisak abban az értelemben, hogy valamennyi pont képe függ az összes (mind az  $n$  darab) kontrollponttól.
- A transzformáció kiszámításának időigénye egyenesen arányos a kijelölt pontpárok számával.
- A transzformációkban a szummák előtti részek affin transzformációkat írnak le, melyek transzformációs mátrixa (homogén koordinátákkal):

$$\begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} a_x & a_y & a_z & a \\ b_x & b_y & b_z & b \\ c_x & c_y & c_z & c \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

így tehát a nem-lineáris deformációkat a szummás tag elemei biztosítják.

A TPS valóban interpoláció, mivel 2-dimenzióban

$$(f_x(Q_i), f_y(Q_i)) = P_i$$



7.20. ábra. Példa TPS interpolációra, ahol a forrás képen (bal) jelöltük ki a  $Q_i$  kontrollpontokat, a  $P_i$  kontrollpontokat pedig a középső képen adtuk meg. A eredményül kapott cél kép (jobb) a töklámpás „arcvonalasait” vette át.

és a 3-dimenziós transzformációnál

$$(f_x(Q_i), f_y(Q_i), f_z(Q_i)) = P_i,$$

vagyis a transzformációk a  $Q_i$  kontrollpontokat a velük párbáállított  $P_i$  kontrollpontokba viszik ( $i = 1, 2, \dots, n$ ).

A TPS interpolációt a 7.19. és a 7.20. ábrák példáival illusztráljuk.

### 7.3. Morphing

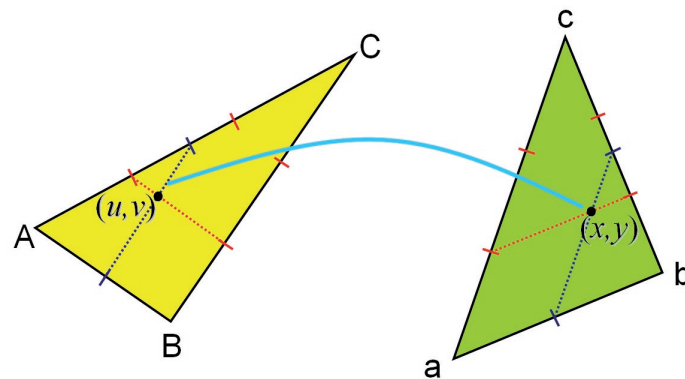
A *morphing* (magyarul metamorfózis, átalakulás, átváltozás) a képfeldolgozásban a forrás kép animált átúsztatását jelenti a cél képbe, mely a *warping* deformációjának végrehajtásán túl még a két kép intenzitásainak összevegyítését is jelenti (az animáció kezdetén a forrás kép intenzitásai dominálnak, majd fokozatosan a cél kép intenzitásai kapnak nagyobb súlyt). Az intenzitások átváltozása a *kereszt-áttűnés* (*cross-dissolving*). Morphing-ra számos példával találkozhattunk már (pl. a *Terminator 2* vagy a *Mask* filmopuszokban). Korábban a morphing költséges és hosszadalmas eljárás volt, míg napjainkra házilag is megvalósítható számos shareware csomaggal. Az alábbiakban a morphing legfontosabb technikáit vázoljuk.

#### Morphing térbeli particionálással

Számos eljárás felbontja a forrás képet és a cél képet egymásnak megfelelőített partíciókra (többnyire háromszögekre vagy négyszögekre). Az animációnál a forrás- és a cél-idomok között interpolációval határozzák meg a közbülső filmkockák felbontásait. Két szomszédos filmkocka egymásnak megfelelőített partícióinak pontjai között a kapcsolatot az alábbi egyszerű affin transzformációval írható le:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

A fenti transzformációs mátrix hat szabad paramétere könnyen meghatározható a forrás- és a cél háromszögek csúcspontjai koordinátáinak behelyettesítésével (ld. 7.21. ábra).



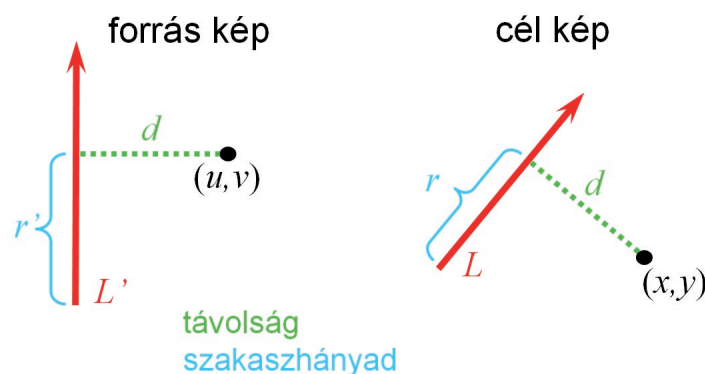
7.21. ábra. A forrás háromszög (bal) és a cél háromszög (jobb) közötti kapcsolat affin transzformációval.

### Kontrollpont-alapú morphing

Kontrollpontok megadásával két technika is kínálkozik: az egymásnak megfeleltetett kontrollpontokból egy nemlineáris transzformációt (pl. TPS) származtathatunk a warping leírására, vagy a kontrollpontok alapján particionáljuk a képtereket (pl. a Delaunay felbontás szerint), majd a morphing valamely térbeli particionálást követő módszerével oldjuk meg a problémát.

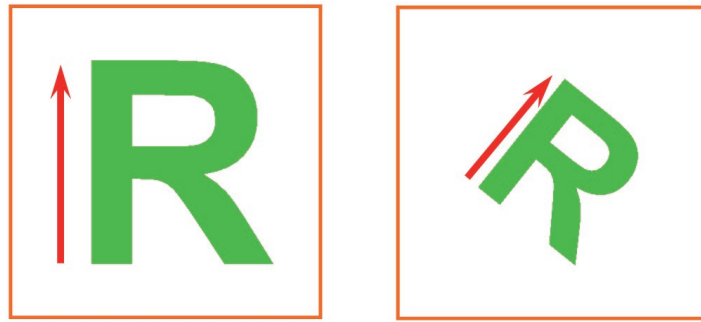
### Mező-alapú morphing

A *mező-alapú* (fields-based) módszer irányított kontroll-szakaszokkal vezérli a deformációt (ld. 7.22. ábra). Egyetlen szakaszpárral egy hasonlósági transzformáció írható le, ahol megengedett az eltolás, az elforgatás és az uniform skálázás (ld. 7.23. ábra).



7.22. ábra. A forrás kép (bal) és a cél kép (jobb) egymásnak megfeleltetett szakaszpárja.

A cél kép  $(x, y)$  pontja transzformációjakor minden egyes  $(L_i, L'_i)$  szakaszpárra meghatározzuk a forrás képen az  $(u_i, v_i)$  ősöket (mintha csak egyetlen szakaszpárunk lenne) és a  $d_i = (x - u_i, y - v_i)$  elmozdulások alapján számítjuk a  $w_i$  súlyokat. Végül az  $(u, v)$  őst az elmozdulások súlyozott átlagából származtatjuk.



7.23. ábra. Példa szakaspárral megadott hasonlósági transzformációra.

A  $w_i$  súlyt az alábbi formula adja:

$$w_i = \left( \frac{l_i^c}{a + d_i} \right)^b$$

ahol:

- $l_i$  az  $L_i$  szakasz hossza,
- $d_i$  az  $(x, y)$  pont és  $L_i$  távolsága (0, ha a pont a szakasz „túloldalán” helyezkedik el),
- $a$ ,  $b$  és  $c$  pedig a vonalak hatását szabályozó konstansok. Az  $a$  paraméter kis értéke is segít elkerülni a nullával való osztást. Növelésével a leképezés simább lesz. A  $b$  paraméter határozza meg az adott szakasz fontosságát a távolság növekedésével. Javasolt értéke 0.5 és 2.0 közötti (0 értéke mellett a szakaszok egyforma súllyal esnek latba). A  $c$  paraméter biztosítja, hogy a hosszabb szakaszok fontosabbak legyenek. Javasolt értéke 0.0 és 1.0 közötti (0 értéke mellett a szakaszok egyforma súllyal bírnak.)

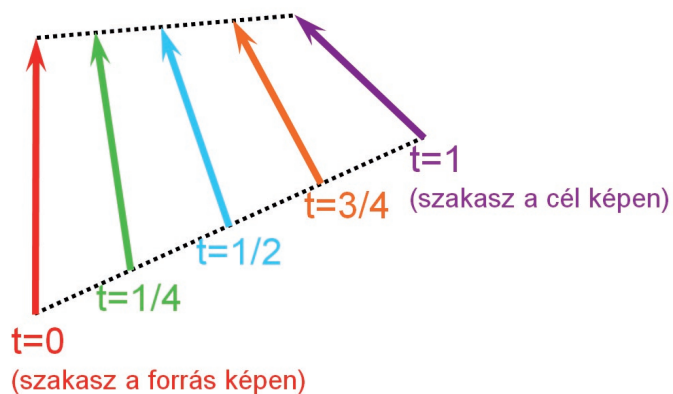
**Algoritmus:** mező-alapú morphing

```

for a cél kép minden  $(x, y)$  pontjára do
   $dist\_sum = (0,0)$ 
   $weight\_sum = 0$ 
  for a minden  $(L_i, L'_i)$  szakaspárra do
     $l_i$  és  $d_i$  számítása
     $(u_i, v_i)$  és  $D_i = (x - u_i, y - v_i)$  számítása
     $w_i$  számítása
     $dist\_sum = dist\_sum + D_i$ 
     $weight\_sum = weight\_sum + w_i$ 
   $(u, v) = (x, y) + dist\_sum / weight\_sum$ 

```

Az animáció során a forrás kép szakaszai és a cél kép szakaszai között (a képkockák számától függően) interpolálunk (ld. 7.24. ábra).



7.24. ábra. A szakaszok interpolációja.

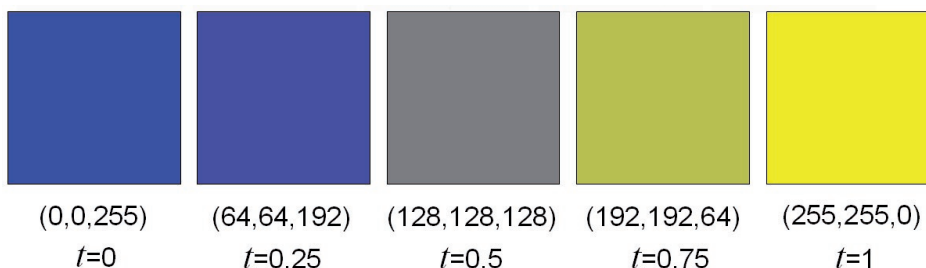
### Kereszt-áttűnés

A kereszt-áttűnés a morphing eljárásoknak a képintenzitásokat meghatározó komponense.

Az animált átalakítás során a forrás kép  $s$  intenzitása fokozatosan átváltozik a célkép  $d$  intenzitásává. A  $t$  időpontban az adott képpont  $i(t)$  intenzitása a következő formula szerint számítandó:

$$i(t) = (1-t) \cdot s + t \cdot d \quad (0 \leq t \leq 1).$$

Kereszt-áttűnésre a 7.25. ábra mutat példát.



7.25. ábra. Példa kereszt-áttűnésre. A „tishta” kék színből négy lépésben jutunk el a „tishta” sárgáig (az RGB kocka egyik testátlóján haladva).

Végezetül a 7.26. ábrával (map\_morphing.eps) illusztráljuk a morphing és a kereszt áttűnés menetét.



7.26. ábra. Példa kereszt-áttűnésre. Az öt kép kockáinak vetítési ideje rendre: 0.00, 0.25, 0.50, 1.75, 1.00.



## 8. fejezet

# Képregisztráció

A képregisztráció a digitális képfeldolgozás egyik alaproblémája. A regisztráció ez esetben nem lajstromba vételt jelent, hanem különböző adathalmazok transzformációját egy közös koordináta rendszerbe.

A jelen fejezet első pontjában áttekintjük a regisztrációt, az arra épülő képillesztés és képfúzió feladatát, valamint a regisztrációs probléma megoldására javasolt módszereket. A második pontban a kontrollpont-párokon alapuló technikákat mutatjuk be, majd a harmadikban az automatikus képregisztráció leghatékonyabb hasonlósági mértékével, a kölcsönös információval foglalkozunk.

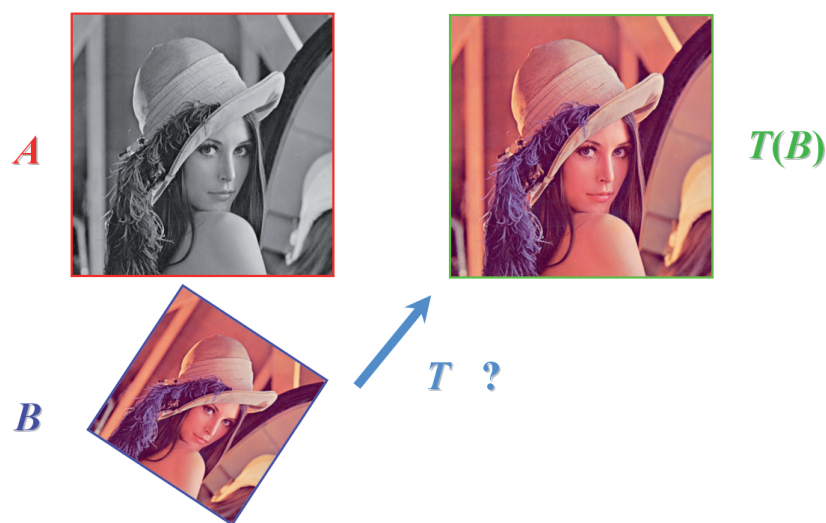
### 8.1. A regisztráció és módszerek

A képfeldolgozás számos területén szükséges különböző nézőpontból, különböző időpontban, vagy akár különböző képkalkító berendezésekkel készült képek közötti geometriai kapcsolat megtalálása, fedésbe hozása egymással. A fentiekkel kapcsolatban három egymásra épülő feladatot különböztethetünk meg.

- A *képregisztráció (image registration)* két kép, az *A* bázis- vagy *referencia-kép* és a *B illesztendő-kép* közötti „legjobb” átfedést, illeszkedést biztosító  $T$  geometriai transzformációt határozza meg.
- A *képillesztés (image matching)* feladata a regisztrációval megtalált transzformáció alkalmazása a *B* illesztendő képre, vagyis a  $C = T(B)$  *illesztett kép* előállítása.
- A *képfúzió (image fusion)* alkalmazásával az *A* referencia- és a *C* illesztett képből egy olyan új *D* képet állítunk elő, amelyen vizsgálhatók a kiindulási képek eltérései, illetve kiegészítik egymást a képtartalmak.

A társproblémákat a 8.1. és a 8.2. ábrákkal szemléltetjük.

A fenti problémák megoldása szükséges egy *kép-mozaik (image mosaic)* előállításához, vagyis egymást részben átfedő kisebb képekből egyetlen nagyobb kialakítása. Mozaik-képekre két példát említünk: A NASA több száz (infravörös tartományban készült) műholdfelvétel összeillesztésével állította elő híres képét, mely a bolygónkat éjszakai kivilágításban

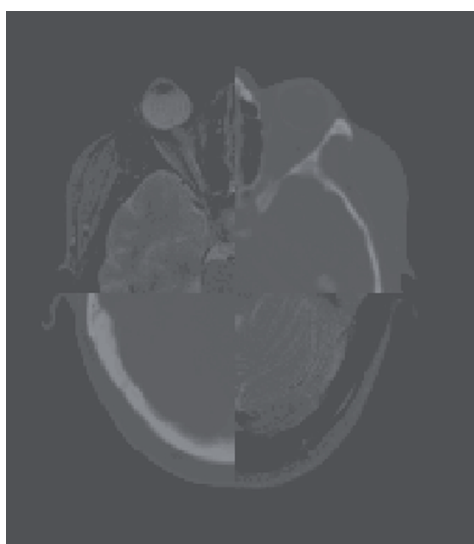


8.1. ábra. Példa képregisztrációra és képilestésre. A regisztráció az **A** és a **B** képek közötti „legjobb” illeszkedést biztosító  $T$  geometriai transzformációt keresi. A képilestés végre is hajtja a transzformációt, vagyis előállítja a  $T(B)$  képet.

mutatja be. Egy, a képet tartalmazó oldal címe:

[http://visibleearth.nasa.gov/view\\_rec.php?id=1438](http://visibleearth.nasa.gov/view_rec.php?id=1438).

A második példa kevésbé lélekemelő: a 2010. októberében bekövetkezett vörösiszap-



8.2. ábra. Példa képfúzióra. Regisztráció és illesztés után egy MR és egy CT agyvizsgálat (transzverzális) szeletének együttes megjelenítése. A képfúzió eredményeként kapott kép bal felső és jobb alsó negyedébe az MR, a másik kettőbe pedig a CT kép került. Hasonló „sakkáblaszerű” megjelenítéshez jutunk az egyes mezők méret-csökkentésével. Egy lehetséges fúzió az is, ha az eredménykép páratlan indexösszegű képpontjainak intenzitásait az első képről, a páratlan indexösszegűekét pedig a másodiktól vesszük.

katasztrófa által érintett területről a Greenpeace megbízásából készült egy 300 nagyfelbontású légifotóból összeszerkesztett kép. A kép megtekinthető a következő címen:

<http://nol.hu/media/file/attach/71/09/00/000000971-2114.jpg>.

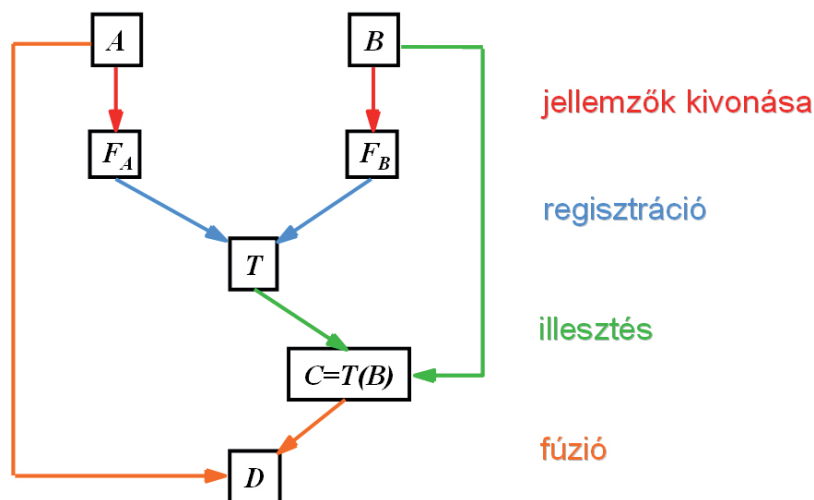
Képek fúziójára is említünk egy példát. A

[www.spacetelescope.org/news/html/heic0602.html](http://www.spacetelescope.org/news/html/heic0602.html)

oldalon a Hubble űrteleszkóppal 1994 és 2003 között készült 51 felvétel fúzióját csodálhatjuk meg a 25 millió fényév távolságra lévő és 170 ezer fényév átmérőjű M101 spirál-galaxisról.

A regisztrációs probléma nem csak képek között merülhet fel. A geometriai kapcsolatot két koordináta rendszer között is lehet keresni, ahol pl. az egyik lehet egy 3-dimenziós orvosi képé, a másik pedig a műtő koordináta-rendszere. Kereshetjük a kapcsolatot egy kép és egy geometriai modell között is.

A három társprobléma megoldását és egymásra épülését a 8.3. ábrával illusztráljuk.



8.3. ábra. A képregisztráció, a képillesztés és a képfúzió egymásra épülése. A regisztrációs probléma megoldása általában (egy előfeldolgozó lépés során) az A és a B képekből kinyert  $F_A$  és  $F_B$  jellemzőkön alapul.

A regisztrációs, képfúziós technikákat számos területen alkalmazzák. Közülük az egyik legfontosabb az orvosi képfeldolgozás. A különböző időpontokban készített felvételek illesztésével lehetőség van pl. egy daganat méretváltozásának megfigyelésére, a különböző képalakító berendezések képeinek fúziójával pedig pl. az anatómiai és a funkcionális képek együttes megjelenítése pontosabb diagnosztikai információt biztosít, illetve segíti a műtét és a terápia tervezését. Műtétevégrehajtás közben a betegről készült kép és a műtőeszköz modelljének együttes megjelenítésével az eljárás vezérelhető illetve ellenőrizhető. Regisztrációval anatómiai atlasz is készíthető, továbbá egy beteg felvétele is összevethető az atlasszal. A regisztráció orvosi alkalmazásairól több áttekintő mű született [32, 37, 39, 43].

A képregisztráció kiemelten fontos a távérzékelésben, a légi- és műholdfelvételek feldolgozásakor (geológia, urbanisztika, célazonosítás), továbbá a számítógépes látás számos területén (pl. mélységi információ számítása sztereoképekből, képszegmentálás, mozgáselemzés) is [14].

A sokrétű regisztrációs problémákra javasolt változatos regisztrációs technikák mindegyike jellemezhető a következő négy komponens megválasztásával [14]:

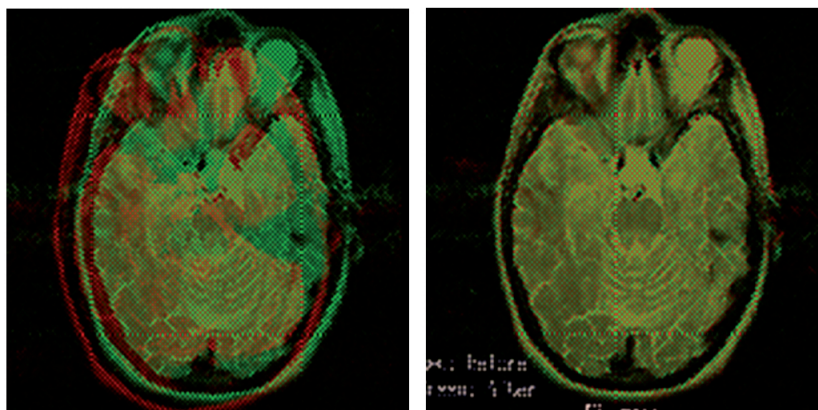
- A *keresési tér* (*search space*) a keresett geometriai transzformáció típusától függ. A keresési tér dimenziója a kérdéses transzformáció szabad paramétereinek számával egyezik meg. Minden egyes transzformációnak (az adott típuson belül) megfelel a keresési tér egy pontja és a regisztrációs probléma megoldása annak a pontnak a megkeresése, amellyel az illesztés „optimális”.
- A *jellemzők tere* (*feature space*) adja meg azt, hogy a kereséshez milyen jellemzőket használunk fel. A jellemzők alkalmas megválasztásával a feldolgozandó, kiértékelendő adatmennyiség radikálisan csökkenthető, így felgyorsítható a keresés. Természetesen szükséges az, hogy a választott jellemzők relevánsak legyenek.
- A *hasonlósági mérték* (*similarity measure*) a keresési téren értelmezett olyan valós függvény, mely (kvantitatív módon) megadja egy ponthoz tartozó transzformáció „jó-ságát”, vagyis azt, hogy az adott transzformációval mennyire „sikerült” illeszteni a bemeneti adatokat. A regisztrációs probléma ennek a függvénynek a (globális) maximum- vagy minimum-helyének a meghatározása.
- A rengeteg (olykor végtelen számú) pontot tartalmazó keresési tér optimális pontjának meghatározása hasonló ahhoz, mint tüt keresni a szénakazalban, így számos eljárásnál szükséges egy hatékony *keresési stratégia* (*searching strategy*) a hasonlósági mérték globális optimumának meghatározására.

Az előző (a 7.) fejezetben már áttekintettük a legfontosabb geometriai transzformációkat és az azok paraméterszámát (a keresési tér dimenzióját).

A képjellemzők, a kiértékelendő adatmennyiség csökkentése az 1990-es évekig a tárolókapacitás szükségessége és az akkori processzorok viszonylag gyenge teljesítménye miatt fontos szerepet játszottak. Akkoriban a nagyméretű (pl. a 3-dimenziós orvosi képek) regisztrációjakor a képek közvetlen illesztése rendkívül időigényes volt. Később a képeket és a bonyolult geometriai modelleket közvetlenül felhasználó módszerek is elterjedtek.

A képi jellemzők lehetnek *külsők* (*extrinsic*) vagy *belső* (*intrinsic*). A külső jellemzők mesterségesen kerülnek a képre, többnyire kifejezetten a regisztrációs probléma megoldásának segítésére (ilyenek pl. orvosi képeken a csontba fűrt, vagy bőrön elhelyezett markerek), míg a belső képjellemzők magából a képtartalomtól származnak (pl. pontok, sarkok, kontúrvonalak, felszínek, objektumok, vagy akár maga a teljes kép). A külső jellemzők hátránya az, hogy egyrészt csak addig lehetséges velük a regisztráció, amíg jelen vannak (pl. egy sztereotaktikus keret eltávolítása után, néhány hónap múlva már újra nem lehet pontosan ugyanúgy rögzíteni és azáltal pontos regisztrációhoz jutni), másrészt pedig orvosi képek esetén nem tekinthetők betegbarát megoldásnak.

A hasonlósági mértékkel szemben támasztott legfontosabb kritérium az, hogy a globális optimumát a legjobb illesztést jelentő transzformációra érje el. (Ha a függvénynek lokális optimumai is léteznek, akkor a keresési stratégiának kell biztosítani azt, hogy azokban ne akadjon el az eljárás.) Kinyert geometriai képjellemzők esetén (pontok, határvonalak, felszínek) azok



8.4. ábra. Példa egymodalitásos regisztrációra. A két 3-dimenziós MR agyvizsgálat ugyanazon transzverzális szeleteinek fuzionált képe regisztráció előtt (bal) és regisztráció után (jobb). A referencia képet piros, az illesztendő és az illesztett pedig zöld színrétegben jelenítettük meg „sakktablaszerűen”.

valamilyen távolság-fogalmon alapuló illesztése a leggyakoribb megközelítés. Ilyen távolság-definíció lehet például az euklideszi legkisebb négyzetes eltérés, a Hausdorff, vagy a chamfer távolság. A hasonlósági mértékek belső képjellemzők esetén lehetnek intenzitás-alapúak (pl. kereszt-korreláció, eltérések négyzetösszege, a különbségek abszolút értékének összege, az előjelváltások száma a különbségképen) vagy együttes előfordulási mátrixon alapulóak (pl. együttes entrópia, kölcsönös információ).

A keresési stratégia célja egy függvény (jelesül a hasonlósági mérték) optimalizálása, a globális szélsőértékének a megtalálása. A függvény-optimalizálás a képregisztrációnál jóval tágabb terület, amelynek ezúttal még a vázlatos összefoglalását sem kíséreljük meg.

Kinyert geometriai jellemzők esetén az optimális transzformáció gyakran egy direkt módszerrel (pl. egy egyenletrendszer megoldásával) közvetlenül is megkapható. Bonyolultabb esetekben pl. iterációs módszerek, dinamikus programozási technikák, illetve a *durvától a finomig* (*coarse-to-fine*) többfelbontású kereső módszerek is alkalmazhatók.

A regisztrációs problémák és technikák számos szempont szerint osztályozhatók, közülük néhány:

- **dimenzió:** lehet 2-dimenziós képet 2-dimenzióssal illeszteni, lehet egy 2-dimenziós képpel legjobban fedésbe kerülő (ferde) 2-dimenziós szeletet megkeresni egy 3-dimenziós képen, lehet 3-dimenziós képet 3-dimenzióssal illeszteni,...;
- **jellemző:** külső vagy belső;
- **interakció:** manuális (a felhasználó az áttetsző illesztendő kép mozgatásával keresi a legjobb/megfelelő illesztést), interaktív (pl. a felhasználó által kijelölt, megadott jellemzőkön alapul a regisztráció), automatikus (egyáltalán nem igényel felhasználói beavatkozást);
- **modalitás:** egymodalitásos (*monomodal*), amikor a képek ugyanolyan berendezéssel készültek (pl. MR-MR, anatómiai kép anatómiai képpel történő illesztése, ahol a képek lehetnek időbeli sorozat elemei (ld. 8.4. ábra), eltérő protokoll-beállításúak, vagy



egy felépítendő atlasz elemei), *többmodalitásos (multimodal)*, amikor a képek különböző elveken alapuló képalkotó berendezésekből származnak (pl. MR-PET, anatómiai információ összevetése funkcionálissal), modalitás illesztése atlasszal, modalitás illesztése objektummal (pl. kép illesztése a műtő koordináta-rendszerével);

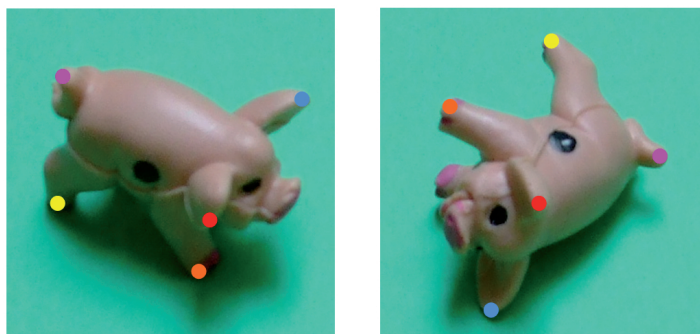
- **forrás:** a regisztráció történhet ugyanazon személy/tárgy képei között (*intrasubject*), különböző személyek/tárgyak képei között (*intersubject*), kép és modell között, vagy kép és atlasz között.

A következő két alfejezet az egymásnak megfeleltetett pont-párokon és a kölcsönös információon alapuló regisztrációs módszerekbe nyújt betekintést.

## 8.2. Pontpárokon alapuló módszerek

A regisztrációra általános megoldást kínálnak a kijelölt pontpárokon mint képi jellemzőkön alapuló módszerek. Egy pont-alapú eljárás a következő három lépésből áll:

- a kontroll-pontok meghatározása, kijelölése a referencia és az illesztendő képen,
- a két képen az egymásnak megfelelő pontok párbaállítása,
- a pontpárok segítségével legjobb illesztést biztosító transzformáció meghatározása.

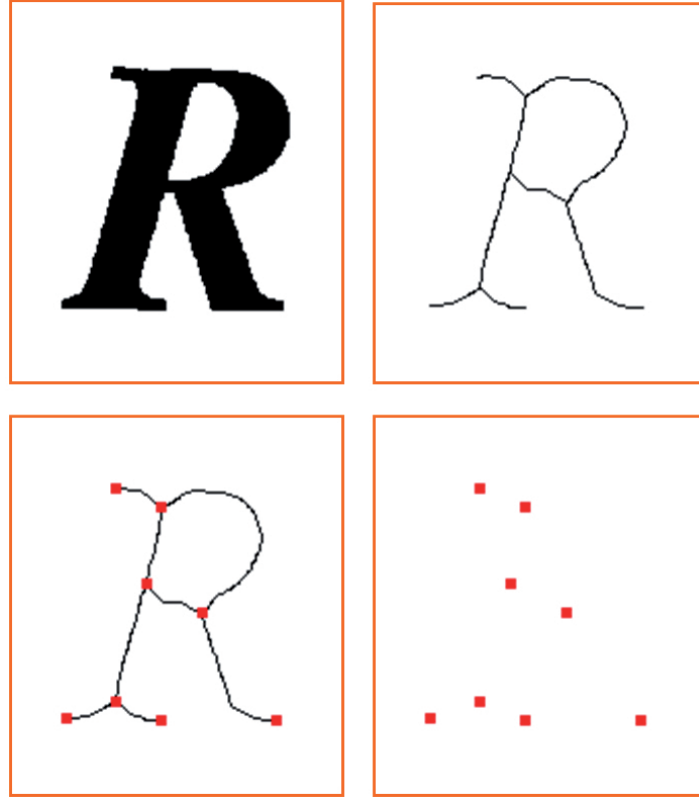


8.5. ábra. Példa kontroll-pontok interaktív kijelölésére és párosítására a „Pass the Pigs” társasjáték „dobókockáin”.

A kontroll-pontok kiválasztását és párosítását elvégezheti a felhasználó is (interaktív módszereknél), vagy történhet automatikusan is. A 8.5. ábra interaktívan kijelölt és párosított pontokra mutat példát, míg a 8.6. ábrán a kontroll-pontok kiválasztása automatikusan történt.

Interaktív kijelöléssel általában kevés (4-20) pontpárt jelölnek ki. A módszer különösen 3-dimenzióban állítja nehéz feladat elé a felhasználót, akinek ráadásul egy alkalmas jó kép-megjelenítőre is szüksége van. Orvosi képek esetében az interaktív kijelölés a felhasználótól nem csak időt, hanem anatómiai ismereteket is igényel. Az automatikus kijelölések nagy számú pontot eredményeznek, melyek párosítása speciális módszerekre, a *megfeleltetési probléma (correspondence problem)* megoldására is szükség van. A megfeleltetési problémával itt most nem foglalkozunk, feltételezzük, hogy a kontroll-pontok párosítása már megtörtént.





8.6. ábra. Példa kontroll-pontok automatikus meghatározására. Egy bináris objektum (fent, bal) vékonyítással meghatározott (ld. 4. fejezet) középvonalán (fent, jobb) könnyen megtalálhatók az elágazási- és a végpontok (lent) a vázpontok osztályozásával (a vázpont szomszédok számának alapján).

A  $k$ -dimenziós euklideszi térben az egymásnak megfeleltetett két ponthalmazt jelölje  $P = \{p_i\}_{i=1,2,\dots,n}$  és  $Q = \{q_i\}_{i=1,2,\dots,n}$ , ahol  $p_i = (p_{i1}, \dots, p_{ik})$  és  $q_i = (q_{i1}, \dots, q_{ik})$ . A regisztráció feladata annak a  $\mathcal{T} : \mathbb{R}^k \rightarrow \mathbb{R}^k$  geometriai transzformációnak a megkeresése, amely a  $Q$  ponthalmazt a  $P$  ponthalmazba képezi le. Amennyiben nem interpolációs transzformációt keresünk, úgy a pontok kijelölési hibája, illetve a vizsgált objektum változása miatt általában a pontos illesztés nem érhető el. Ezért az optimális  $\mathcal{T}$  transzformációtól azt várjuk el, hogy a

$$\psi(\mathcal{T}) = \sum_{i=1}^n \|T(q_i) - p_i\|^2$$

hibafüggvény, hasonlósági mérték minimális legyen.

A következő alfejezetekben a leggyakrabban használt transzformációkra, keresési terekre vázoljuk fel a pont-alapú módszereket. A terület iránt behatóbban érdeklődők figyelmébe Tanács áttekintő munkáját ajánljuk [69]. Valamennyi módszerben közös, hogy a hasonlósági mérték optimalizálása közvetlenül, egyenletrendszerek megoldásával történik, vagyis direkt keresési stratégiát követnek.

Fontos megjegyezni, hogy egy adott típusú deformációt legjobban modellező transzformációt célszerű keresni és nem valamely nagyobb szabadságfokú transzformáció módszerét,

mivel csak a kontroll-pontok pontos kijelölésekor kapunk alacsonyabb szabadságfokú transzformációt a magasabb szabadságfokú speciális eseteként.

A pont-alapú módszereknél ajánlott minél több pontpárt, minél pontosabban és minél nagyobb térrészt kifeszítve kijelölni.

### 8.2.1. Merev-test transzformáció keresése

A merev-test transzformációnál csak eltolás és elforgatás engedélyezett, vagyis távolság-, párhuzamosság- és szögtartó. A hasonlósági transzformációnál az eltolás és elforgatás mellett még az uniform skálázás is megengedett.

A pont-alapú merev-test transzformáció meghatározására Arun és munkatársai közöltek megoldást [1], amit Umeyama terjesztett ki hasonlósági transzformáció keresésére [73].

Arun és munkatársai a transzformációt az alábbi alakban keresték:

$$\psi = \sum_{i=1}^n \|p_i^T - (s \cdot R \cdot q_i^T + T)\|^2,$$

ahol  $R$  egy  $k \times k$  méretű forgató mátrix (vagyis eleget tesz az  $R \cdot R^T = I_m$  és a  $\det(R) = 1$  feltételeknek, ahol  $R^T$  az  $R$  transzponáltja,  $I_m$  pedig a  $k \times k$  méretű egységmátrix),  $T$  egy  $k$ -elemű vektor, amely az eltolást adja meg,  $s$  pedig a (minden tengely mentén egyforma) skálázási tényező. Ha merev-test transzformációra van szükségünk, akkor  $s=1$ . Az algoritmus lépései:

Számítsuk ki a két ponthalmaz súlypontját:

$$\mu_p = \frac{1}{n} \sum_{i=1}^n p_i^T, \quad \mu_q = \frac{1}{n} \sum_{i=1}^n q_i^T,$$

és legyen

$$d_i = q_i - \mu_q, \quad d'_i = p_i - \mu_p, \quad \sigma_q^2 = \sum_{i=1}^n \|q_i - \mu_q\|^2,$$

$$H = \sum_{i=1}^n d_i \cdot d_i'^T.$$

Legyen a  $k \times k$  méretű  $H$  mátrix szinguláris érték szerinti felbontása  $H = U \cdot D \cdot V^T$  [50], és legyen  $X = V \cdot U^T$ . Ha  $\det(X)=1$ , akkor legyen  $\bar{R}=X$  az optimális forgató mátrix,  $\bar{s}=\sigma_q^2/\text{tr}(D)$  pedig az optimális skálázás. Ha  $\det(X)=-1$ , akkor  $X$  tükrözést ad. Mivel ez utóbbi számunkra nem megfelelő, ezért ebben az esetben további vizsgálatra van szükség. Ha  $H$  szinguláris értékei közül egy 0, akkor  $\bar{R} = V' \cdot U^T$ , és  $\bar{s} = \sigma_q^2/\text{tr}(D')$ , ahol a  $V'$  és  $D'$  mátrixokat úgy kapjuk, hogy a  $V$  és  $D$  mátrixok utolsó oszlopbeli elemei előjelét megfordítjuk. Ha egyik szinguláris érték sem 0, akkor nincs megoldás.

Az optimális forgató mátrix és skálázás ismeretében az optimális  $\bar{T}$  eltolási vektor megkapható  $\bar{T} = \mu_p - \bar{s} \cdot \bar{R} \cdot \mu_q$  alakban.

### 8.2.2. Affin transzformáció keresése

Az affin transzformáció az eltolás, az elforgatás és a skálázás mellett a nyírást is megengedi. A lineáris transzformáció megőrzi a párhuzamosságot, de nem szögtartó.

Az  $\mathcal{T} : \mathbb{R}^k \rightarrow \mathbb{R}^k$   $k$ -dimenziós ( $k = 2, 3, \dots$ ) affin transzformáció megadható egy

$$T = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1k} & t_{1,k+1} \\ t_{21} & t_{22} & \cdots & t_{2k} & t_{2,k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{k1} & t_{k2} & \cdots & t_{kk} & t_{k,k+1} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

$(k+1) \times (k+1)$  méretű (homogén koordinátás [23]) valós mátrixszal. Tetszőleges  $x = (x_1, \dots, x_k)$  és  $y = (y_1, \dots, y_k) \in \mathbb{R}^k$  esetén  $y = \mathcal{T}(x)$  akkor és csak akkor, ha

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \\ 1 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1k} & t_{1,k+1} \\ t_{21} & t_{22} & \cdots & t_{2k} & t_{2,k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{k1} & t_{k2} & \cdots & t_{kk} & t_{k,k+1} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ 1 \end{pmatrix}.$$

Keressük közvetlenül a transzformáció mátrixának ismeretlen elemeit. Legyenek  $P$  és  $Q$  az előzőekben definiált pontthalmazok, a  $\{p'_i\}_{i=1,2,\dots,n}$  pedig  $\{q_i\}$   $\mathcal{T}$  transzformáció melletti képe, vagyis  $p'_i = \mathcal{T}(q_i)$  ( $i = 1, 2, \dots, n$ ). Ekkor a hibavektor euklideszi normáját az alábbi alakban írhatjuk fel:

$$\psi(t_{11}, \dots, t_{k,k+1}) = \sum_{i=1}^n \|p'_i - p_i\|^2 = \sum_{i=1}^n \sum_{j=1}^k (t_{j1} \cdot q_{i1} + \dots + t_{jk} \cdot q_{ik} + t_{j,k+1} - p_{ij})^2.$$

A  $\psi$  ott veheti fel a minimumát, ahol a  $\frac{\partial \psi}{\partial t_{11}}, \dots, \frac{\partial \psi}{\partial t_{k,k+1}}$  parciális deriváltak mindegyike egyenlő nullával:

$$\frac{\partial \psi}{\partial t_{uv}} = 2 \cdot \sum_{i=1}^n q_{iv} \cdot (t_{u,k+1} - p_{iu} + \sum_{l=1}^k t_{ul} \cdot q_{il}) = 0$$

$$(u = 1, 2, \dots, k, v = 1, 2, \dots, k),$$

$$\frac{\partial \psi}{\partial t_{u,k+1}} = 2 \cdot \sum_{i=1}^n (t_{u,k+1} - p_{iu} + \sum_{l=1}^k t_{ul} \cdot q_{il}) = 0$$

$$(u = 1, 2, \dots, k).$$

Bevezetve az

$$\begin{aligned} a_{uv} &= a_{vu} = \sum_{i=1}^n q_{iu} \cdot q_{iv}, \\ b_u &= \sum_{i=1}^n q_{iu}, \\ c_{uv} &= \sum_{i=1}^n p_{iu} \cdot q_{iv}, \\ d_u &= \sum_{i=1}^n p_{iu} \end{aligned}$$

$(u = 1, 2, \dots, k, v = 1, 2, \dots, k)$  jelöléseket, a következő  $k \cdot (k+1)$  elemű egyenletrendszert kapjuk:

$$\begin{pmatrix} \begin{matrix} a_{11} & \dots & a_{1k} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k1} & \dots & a_{kk} & b_k \\ b_1 & \dots & b_k & n \end{matrix} & 0 & \begin{pmatrix} t_{11} \\ \vdots \\ t_{1k} \\ t_{1,k+1} \\ t_{21} \\ \vdots \\ t_{2k} \\ t_{2,k+1} \\ \vdots \\ t_{k1} \\ \vdots \\ t_{kk} \\ t_{k,k+1} \end{pmatrix} \\ 0 & \begin{matrix} a_{11} & \dots & a_{1k} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k1} & \dots & a_{kk} & b_k \\ b_1 & \dots & b_k & n \end{matrix} & \begin{pmatrix} c_{11} \\ \vdots \\ c_{1k} \\ d_1 \\ c_{21} \\ \vdots \\ c_{2k} \\ d_2 \\ \vdots \\ c_{k1} \\ \vdots \\ c_{kk} \\ d_k \end{pmatrix} \end{pmatrix} =$$

A fenti egyenletrendszernek akkor és csak akkor létezik egyértelműen a megoldása, ha  $\det(M) \neq 0$ , ahol

$$M = \begin{pmatrix} a_{11} & \dots & a_{1k} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{k1} & \dots & a_{kk} & b_k \\ b_1 & \dots & b_k & n \end{pmatrix},$$

ami ekvivalens azzal, hogy a  $q_i$  pontok nem esnek ugyanazon  $(k-1)$ -dimenziós hipersíkra [68]. Ez azt jelenti, hogy 2-dimenziós esetben a  $Q$  halmaz legalább három, nem egy egyenesbe eső pontot tartalmaz, 3-dimenzióban pedig legalább négy, nem egy síkba eső pontot kell képenként kijelölni.

### 8.2.3. Perspektív transzformáció

A perspektív transzformáció a legáltalánosabb lineáris transzformáció. Egyenes képe egyenes lesz, de sem a távolságot, sem a szöget, sem a párhuzamosságot nem tartja. Használata szükséges lehet például három dimenziós tájról készült 2-dimenziós képek (fényképek, légifotók) illesztésekor.

A transzformáció tetszőleges  $k$ -dimenzióban megadható egy

$$T = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1k} & t_{1,k+1} \\ t_{21} & t_{22} & \cdots & t_{2k} & t_{2,k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{k1} & t_{k2} & \cdots & t_{kk} & t_{k,k+1} \\ t_{k+1,1} & t_{k+1,2} & \cdots & t_{k+1,k} & 1 \end{pmatrix}$$

(homogén koordinátás) valós mátrixszal.

A 2-dimenziós esetet fejtjük most ki részletesebben, magasabb dimenziószám esetén hasonló módon kaphatjuk meg a megoldást. A mátrixszorzás elvégzése és átrendezés után a következő egyenleteket kapjuk tetszőleges  $i$  ( $1 \leq i \leq n$ ) pontpárra:

$$p_{i1} = t_{11} \cdot q_{i1} + t_{12} \cdot q_{i2} + t_{13} - t_{31} \cdot q_{i1} \cdot p_{i1} - t_{32} \cdot q_{i2} \cdot p_{i1}$$

$$p_{i2} = t_{21} \cdot q_{i1} + t_{22} \cdot q_{i2} + t_{23} - t_{31} \cdot q_{i1} \cdot p_{i2} - t_{32} \cdot q_{i2} \cdot p_{i2}.$$

Amennyiben legalább négy, megfelelő helyzetű pontpárunk van, akkor egy lineáris (négy-nél több pontpár esetén túlhatározott) egyenletrendszert kapunk, amelynek a legkisebb négyzetes megoldását a következő egyenletrendszer megoldásával kapjuk:

$$A^T \cdot A \cdot \begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{21} \\ t_{22} \\ t_{23} \\ t_{31} \\ t_{32} \end{pmatrix} = A^T \cdot \begin{pmatrix} p_{11} \\ \vdots \\ p_{n1} \\ p_{12} \\ \vdots \\ p_{n2} \end{pmatrix},$$

ahol

$$A_{(2 \cdot n \times 8)} = \begin{pmatrix} q_{11} & q_{12} & 1 & 0 & 0 & 0 & -q_{11} \cdot p_{11} & -q_{12} \cdot p_{11} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & 1 & 0 & 0 & 0 & -q_{n1} \cdot p_{n1} & -q_{n2} \cdot p_{n1} \\ 0 & 0 & 0 & q_{11} & q_{12} & 1 & -q_{11} \cdot p_{12} & -q_{12} \cdot p_{12} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & q_{n1} & q_{n2} & 1 & -q_{n1} \cdot p_{n2} & -q_{n2} \cdot p_{n2} \end{pmatrix}.$$

### 8.2.4. Polinomiális nemlineáris transzformációk

Bizonyos esetekben a képalkotó berendezés globális nemlineáris torzítást okoz a képen, ilyenkor lineáris transzformáció helyett nemlineárisra van szükség. A legegyszerűbb ilyen transzformációk az  $m$ -edfokú polinomiális transzformációk, amelyekre az a jellemző, hogy egy

egyenes képe egy  $m$ -edfokú polinom által meghatározott görbe lesz [38]. A transzformációt a következő módon definiáljuk 2-dimenzióban:

$$u_x = \sum_{i=0}^m \sum_{j=0}^{m-i} a_{ij} \cdot v_x^i \cdot v_y^j$$

$$u_y = \sum_{i=0}^m \sum_{j=0}^{m-i} b_{ij} \cdot v_x^i \cdot v_y^j$$

$$u = (u_x, u_y) \in \mathbb{R}^2, \quad v = (v_x, v_y) \in \mathbb{R}^2,$$

illetve 3D-ben:

$$u_x = \sum_{i=0}^m \sum_{j=0}^{m-i} \sum_{k=0}^{m-i-j} a_{ijk} \cdot v_x^i \cdot v_y^j \cdot v_z^k$$

$$u_y = \sum_{i=0}^m \sum_{j=0}^{m-i} \sum_{k=0}^{m-i-j} b_{ijk} \cdot v_x^i \cdot v_y^j \cdot v_z^k$$

$$u_z = \sum_{i=0}^m \sum_{j=0}^{m-i} \sum_{k=0}^{m-i-j} c_{ijk} \cdot v_x^i \cdot v_y^j \cdot v_z^k$$

$$u = (u_x, u_y, u_z) \in \mathbb{R}^3, \quad v = (v_x, v_y, v_z) \in \mathbb{R}^3.$$

Látható, hogy az  $m = 1$  esetben az affin transzformációt kapjuk. Gyakorlatban másod- illetve harmadfokú polinomokat használnak. Ennél nagyobb fokszám esetén a transzformáció paramétereinek meghatározása egyre számításigényesebb, és a minimálisan szükséges pontpárok  $T(m)$  száma is négyzetesen nő.

Az egyszerűbb felírási mód kedvéért a másodfokú 2-dimenziós eset legkisebb négyzetes megoldását adjuk meg. Az eljárás könnyen kiterjeszthető magasabb fokszámra és dimenzióra is.

Legyenek  $\{p_i\}_{i=1,2,\dots,n}$  és  $\{q_i\}_{i=1,2,\dots,n}$  az előzőekben definiált ponthalmazok. Az előző alfejezetben megadott képlet kifejtve a 2-dimenziós másodfokú ( $m = 2$ ) esetre:

$$p_{i1} = a_{00} + a_{10} \cdot q_{i1} + a_{01} \cdot q_{i2} + a_{20} \cdot q_{i1}^2 + a_{11} \cdot q_{i1} \cdot q_{i2} + a_{02} \cdot q_{i2}^2$$

$$p_{i2} = b_{00} + b_{10} \cdot q_{i1} + b_{01} \cdot q_{i2} + b_{20} \cdot q_{i1}^2 + b_{11} \cdot q_{i1} \cdot q_{i2} + b_{02} \cdot q_{i2}^2.$$

A legkisebb négyzetes megoldást a következő egyenletrendszerek megoldásával kapjuk:

$$A^T \cdot A \cdot \begin{pmatrix} a_{00} & b_{00} \\ a_{10} & b_{10} \\ a_{01} & b_{01} \\ a_{20} & b_{20} \\ a_{11} & b_{11} \\ a_{02} & b_{02} \end{pmatrix} = A^T \cdot \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ \vdots & \vdots \\ p_{n1} & p_{n2} \end{pmatrix},$$



ahol

$$A = \begin{pmatrix} 1 & q_{11} & q_{12} & q_{11}^2 & q_{11} \cdot q_{12} & q_{12}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & q_{n1} & q_{n2} & q_{n1}^2 & q_{n1} \cdot q_{n2} & q_{n2}^2 \end{pmatrix}.$$

A nemlineáris transzformációk közé tartozik az előző (a 7.) fejezetben tárgyalt vékony fémlemez interpoláció (thin plate spline, TPS) is, amelynek paraméterei szintén egy lineáris egyenletrendszer megoldásával kaphatók meg.

### 8.3. Kölcsönös információn alapuló regisztráció

A regisztráció gyakran egy összetett probléma részeként jelentkezik. Ilyen esetekben fontos, hogy a problémát minél kisebb mértékű felhasználói beavatkozással oldjuk meg, így különösen fontos a megbízható automatikus módszerek kifejlesztése. Az automatikus regisztrációra kétféle megközelítést javasoltak.

Az elsőnél a képekből geometriai információkat (például sarokpontokat, kontúrokat, fel-színeket) nyernek ki és ezek alapján határozzák meg az optimális illesztést. Ekkor külön feladat például a pontok párosítása, illetve annak biztosítása, hogy egymásnak megfeleltethető jellemzők kerüljenek kinyerésre a képekből. Ez a megközelítés jól működik egymodalitásos esetben, vagyis amennyiben a képek ugyanolyan típusú (és beállítású) képalkotó berendezés-ből származnak és a képalkotás körülményei lényegesen nem különböznek. Többmodalitásos esetben viszont sokszor nehéz ezt a célt elérni.

A második megközelítés esetén közvetlenül a képek intenzitásait használjuk fel, jellemzők kinyerése nélkül. A következőkben az intenzitások együttes előfordulási mátrixán alapuló kölcsönös információt (mint hasonlósági mértéket) ismertetjük, mely sikerrel alkalmazható multimodális orvosi regisztrációs problémák megoldására is.

A kölcsönös információt (*mutual information*) hasonlósági mértéket Collignon és munkatársai [18] valamint Viola és Wells [75, 78] javasolták, Studholme és munkatársai pedig a normalizált kölcsönös információt [65] vezette be.

Mindkét mérték felhasználja a képek entrópiáját. Az  $A$  kép entrópiája

$$H(A) = - \sum_a p_A(a) \cdot \log p_A(a),$$

ahol  $p_A(a)$  a képen az „ $a$ ” intenzitás előfordulásának gyakoriságát jelöli, vagyis a kép hisztogramját az  $a$ -adik helyen.

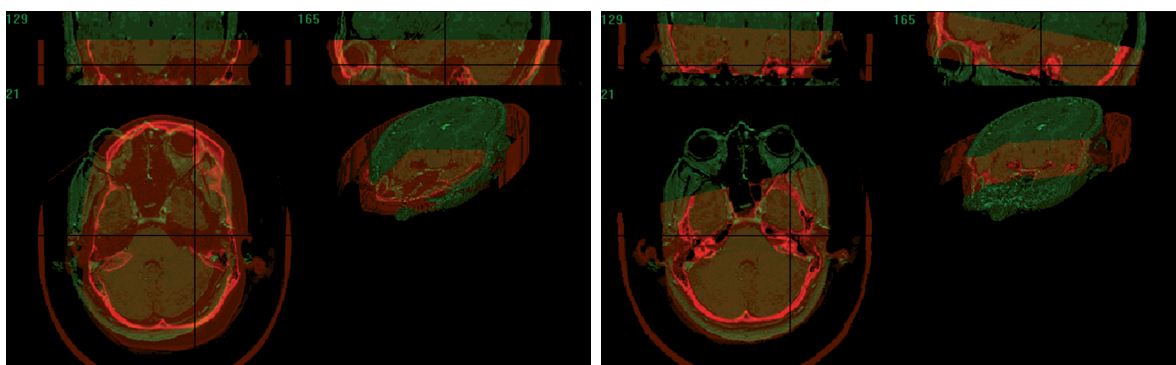
Szükség van még az  $A$  és a  $B$  képek

$$H(A, B) = - \sum_{a,b} p_{AB}(a, b) \cdot \log_2 p_{AB}(a, b)$$

együttes entrópiájára is, ahol  $p_{AB}(a, b)$  jelöli azt, hogy az egymásra helyezett  $A$  és  $B$  képen mekkora gyakorisággal fordul elő az, hogy az  $A$  kép „ $a$ ” intenzitású pontja a  $B$  kép „ $b$ ” intenzitású pontjával kerül fedésbe.

A kölcsönös információ számítása a következő:

$$MI(A, B) = H(A) + H(B) - H(A, B),$$



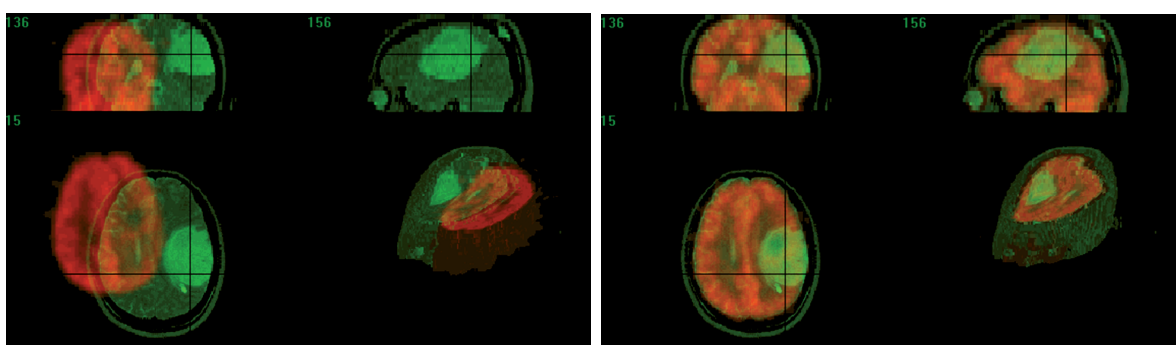
8.7. ábra. Példa kölcsönös információon alapuló multimodális regisztrációra. A referencia 3-dimenziós MR agyvizsgálat piros színrétegben, az illesztendő CT vizsgálat pedig zöldben megjelenített regisztráció előtt (bal) és regisztráció után (jobb). Dacára annak, hogy a CT kép jóval kisebb térfogatot tartalmaz, mint az ugyancsak anatómiai információt hordozó MR, a kölcsönös információon alapuló módszer pontos illeszkedést biztosít.

a normalizált kölcsönös információé pedig:

$$NMI(A, B) = \frac{H(A) + H(B)}{H(A, B)}.$$

A fenti hasonlósági mértékeket a regisztrációs eljárások egy  $T$  transzformáció kiértékelésekor az  $A$  referencia és a  $T(B)$  transzformált illesztendő képre értékelik ki és azt a  $T$  transzformációt keressük, melyre  $MI(A, T(B))$  minimális, vagy az  $NMI(A, T(B))$  maximális.

Tanács a kölcsönös információon alapuló automatikus módszerével merev test transzformációt keresett és az eljárást többfelbontású piramis reprezentációval gyorsította [69]. A módszer eredményességét a 8.7. és a 8.8. ábrákon mutatjuk be.



8.8. ábra. Példa kölcsönös információon alapuló multimodális regisztrációra anatómiai és funkcionális képek között. A referencia 3-dimenziós MR agyvizsgálat zöld, az illesztendő PET vizsgálat pedig piros színrétegben megjelenített regisztráció előtt (bal) és regisztráció után (jobb). A kölcsönös információon alapuló módszer pontos illeszkedést talált az anatómiai MR kép és az anatómiai nyomokat nem hordozó funkcionális PET vizsgálat között.

## 9. fejezet

# Mozgóképek feldolgozása

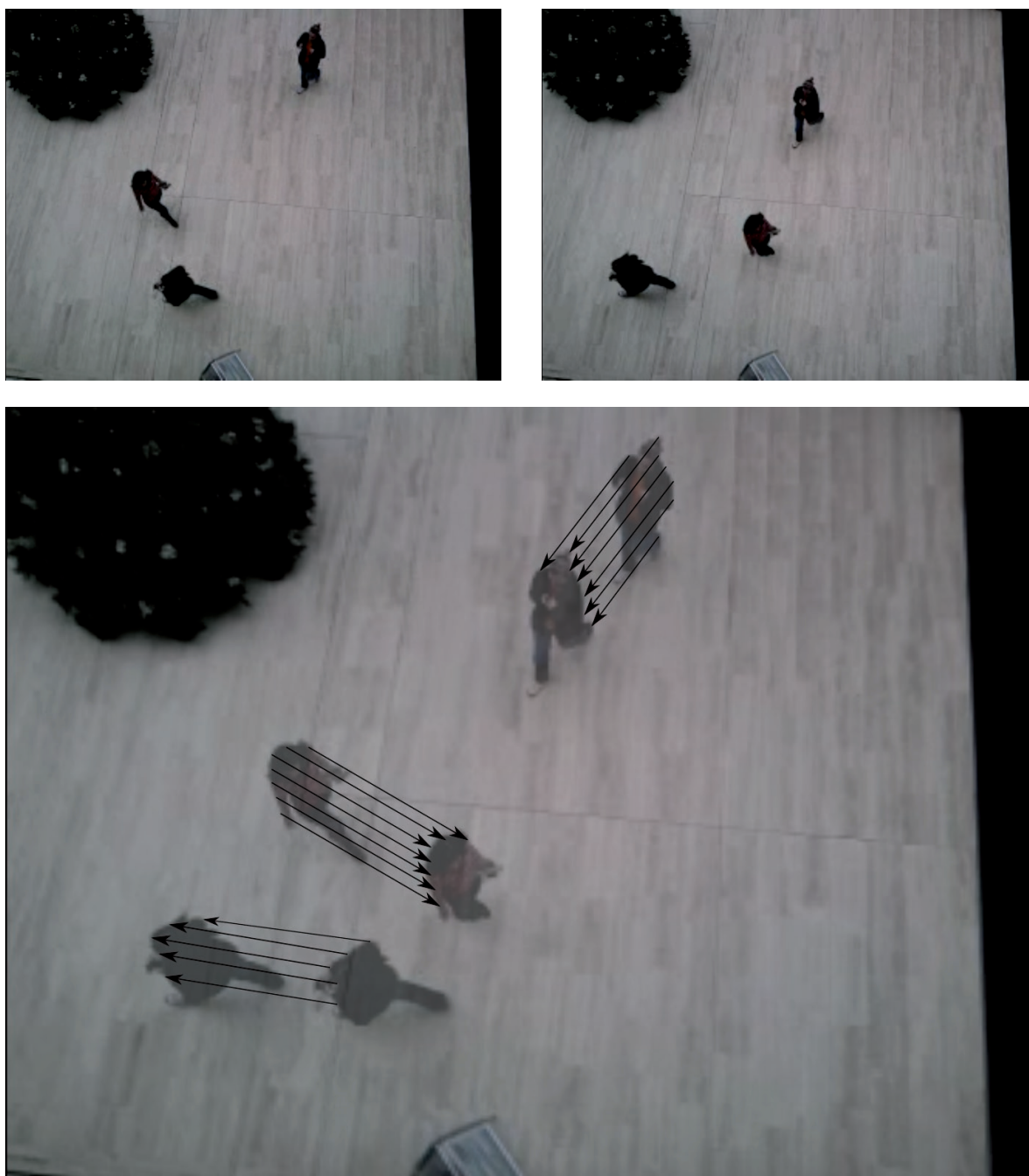
A mozgóképeket alkotó berendezések elterjedésével és a számítógépek teljesítmény-növekedésével a mozgóképek feldolgozására, a *mozgás-elemzésre* (*motion analysis*) egyre nagyobb az igény. A mozgóképek analízisét gyakran valós időben kell megoldani (pl. a robot-navigációnál). A mozgóképek elemzésének ugyancsak fontos alkalmazási területe a 3-dimenziós objektumok alakjának detektálása. A mozgással kapcsolatos problémákat a következő három fő kategóriába sorolhatjuk:

- a mozgás detektálása egyetlen (álló) kamerával,
- objektum-detektálása és objektum helyének meghatározása (mozgó objektumra statikus kamerával vagy mozgó kamera és álló objektum esetén),
- 3-dimenziós tulajdonságok származtatása egy mozgó objektumról különböző időpontokban készített 2-dimenziós projekciók alapján.

A mozgókép-elemzés alkalmazásai sokrétűek: némelyeknél elegendő pusztán azt jelezni, hogy mozgás történt, más alkalmazásoknál pedig a mozgás irányának, sebességének meghatározására, sőt a mozgó tárgyak leírására, felismerésére is igény van. Jóllehet a mozgás-elemzést gyakran dinamikus képanalízisnek nevezik, a módszerei jobbra csak néhány egymást követő képkockán alapulnak és egymásnak megfeleltetett „érdekes” pontpárokon alapulnak. Az 3-dimenziós mozgás 2-dimenziós reprezentációja a *mozgás-mező* (*motion field*), ahol minden egyes ponthoz egy *sebesség-vektort* (*velocity vector*) rendelnek, mely megfelel a mozgás irányának, sebességének és az elmozdulás távolságának. Mozgás-mezőre a 9.1. ábra mutat példát.

A terület feldolgozásánál ugyanabba a problémába ütköztünk, mint a többi fejezetnél: felettébb nehéz egy komplett kurzust is kitölteni képes témakört egyetlen rövid fejezetbe sűríteni. A mozgás-analízis módszerei közül a következő három pontban a különbség-képek elemzését, az optikai áramlás kiszámítását és az egymásnak megfeleltetett érdekes pontokon alapuló technikákat vázoljuk.

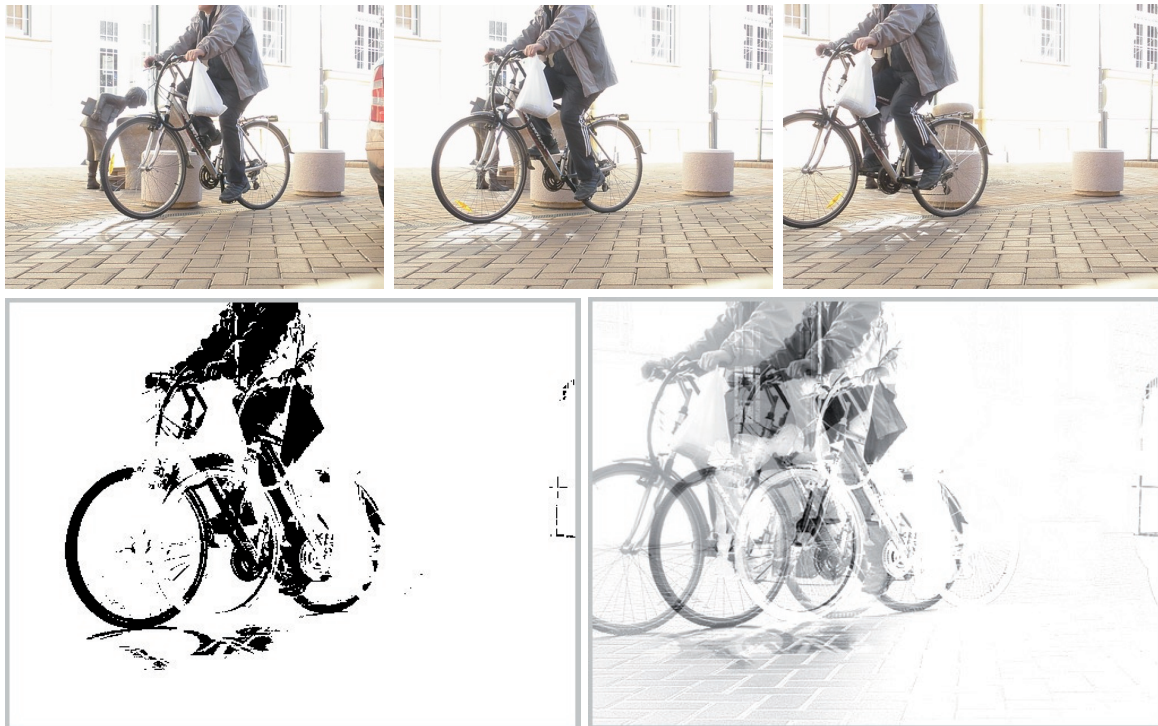
A terület feldolgozását Sonka, Hlavac és Boyle kézikönyvére [64] alapoztuk.



9.1. ábra. Példa mozgás-mezőre. Egy mozgókép két egymást követő kockája (fent) és mozgás-mező néhány vektora a fuzionált képre vetítve (lent).



## 9.1. Különbségképeken alapuló mozgás-analízis



9.2. ábra. Példa különbség-képre és felhalmozott különbség-képre. Egy mozgókép három egymást követő kockája (fent), az első kettő különbség-képe (lent, bal) és a három kockából számított felhalmozott különbség-kép (lent, jobb).

A mozgás (amennyiben a kamera mozdulatlan és a megvilágítás állandó) egyszerűen és hatékonyan detektálható a *különbség-képek* (*difference image*) segítségével. A különbség-kép egy bináris kép két egymást (adott idő-intervallummal) követő képkocka, az  $f_1$  és az  $f_2$  alapján a következőképpen számítunk:

$$d(x, y) = \begin{cases} 0, & \text{ha } |f_1(x, y) - f_2(x, y)| \leq \tau \\ 1, & \text{különben} \end{cases},$$

ahol  $\tau$  a változás-detektálás érzékenységet szabályozó küszöbérték. A különbségkép valamely pontja az alábbi esetekben „1” (változást detektáló):

- $f_1(x, y)$  és  $f_2(x, y)$  közül az egyik egy mozgó objektumhoz, a másik pedig a statikus háttérhez tartozik,
- $f_1(x, y)$  és  $f_2(x, y)$  különböző mozgó objektumok pontjai,
- $f_1(x, y)$  és  $f_2(x, y)$  ugyanazon mozgó objektum különböző (eltérő intenzitású) pontjai, vagy
- kiküszöbölendő rendszerhiba történt (zajjal terhelt valamelyik kép, elmozdult a kamera, vagy megváltozott a megvilágítás).



9.3. ábra. Példa különbség-képre és felhalmozott különbség-képre. Egy mozgókép három egymást követő kockája (fent), az első kettő különbség-képe (lent, bal) és a három kockából számított felhalmozott különbség-kép (lent, jobb).

A módszer hátránya egyrészt az, hogy az alkalmazhatósága erősen függ az objektum-háttér kontrasztjától, másrészt pedig a *röppálya* (*trajectory*), vagyis az elmozdulás iránya rejtve marad a különbség-képeken. Amennyiben a mozgás iránya és más időtől függő jellemző is meghatározandó, úgy a *felhalmozott különbség-kép* (*cumulative difference image*) számítása nyújthat megoldást. A felhalmozott különbség-kép egy  $n$ -elemű képsorozatra a különbségképek súlyozott összege:

$$d_{\text{cum}}(x, y) = \sum_{k=1}^n w_k \cdot |f_1(x, y) - f_k(x, y)|,$$

ahol a  $w_k$  súlyok jelzik az egyes képek fontosságát. Általában az időben közelebbi képek (és a rajtuk történt változások) a fontosabbak, így a súlyok megválasztásakor a  $w_1 \leq w_2 \leq \dots \leq w_n$  relációk állnak fenn.

Különbség-képekre és felhalmozott különbség-képekre a 9.2. és a 9.3. ábrákkal adunk példákat.

A mozgás-mező konstruálásakor több probléma is felléphet. Pl. ha egy téglalapot az egyik oldalán csúsztatunk, akkor a mozgás-mező nem egy objektum elmozdulását mutatja, hanem két mozgó élet. A másik példa az *apertúra probléma* (*aperture problem*). Ha egy „méretes” konstans intenzitású tárgy mozog „kis” ablak előtt, akkor nem egyértelmű a mozgás iránya.

A különbség-képek jelzik a mozgás előfordulát, de a mozgás jellemzőiről nem nyújtanak megbízható információt. Megbízhatóbb jellemzéshez jutunk, ha az  $f_1$  és az  $f_2$  képkockákra



a következő *mozgó él-képet* (*moving edge image*) számítjuk:

$$d_{\text{med}}(x, y) = |\nabla f_1(x, y)| \cdot |f_1(x, y) - f_2(x, y)|,$$

melynél az intenzitásokat és az egymásnak megfelelő területeket is összevetjük. (A szorzat első tagja a  $\nabla f_1(x, y)$  az első képkockán az él magnitúdót jelöli.)

## 9.2. Optikai áramlás

A mozgás-mező egy elméleti konstrukció, mivel a gyakorlatban minden egyes képpontnak egyedi mozgása lehet. A mozgás-mező meghatározására egy gyakran alkalmazott módszer az *optikai áramlás* (*optical flow*) [33] számításán alapul, ahol a képkockák olyan rövid időközönként követik egymást, hogy az egymást követő képkockákon nincs szignifikáns változás.

Az optikai áramlás számítása az alábbi két feltételen alapul:

- A pontok érzékelt fényessége nem változik (konstans az idő függvényében).
- A közeli pontok hasonlóképpen mozognak a kép síkjában.

Tegyük fel, hogy a folytonos képünkre  $f(x, y, t)$  jelöli az  $(x, y)$  pozíció intenzitását a  $t$  időpontban. A függvény Taylor-sorba fejtésével az alábbi egyenlet teljesül:

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) + \Delta x \cdot f_x + \Delta y \cdot f_y + \Delta t \cdot f_t + \mathcal{O}(\delta^2).$$

Feltételezhetjük, hogy az  $(x, y)$  pozíció környezetét a  $(\Delta x, \Delta y)$  „kis” távolsággal eltolva, a „rövid”  $\Delta t$  idő elteltével fennáll az alábbi:

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t).$$

Ha a fenti Taylor sorfejtésben a  $\mathcal{O}(\delta^2)$  hibtag elhanyagolható, akkor a feltételezésünk szerint:

$$\Delta x \cdot f_x + \Delta y \cdot f_y + \Delta t \cdot f_t = 0.$$

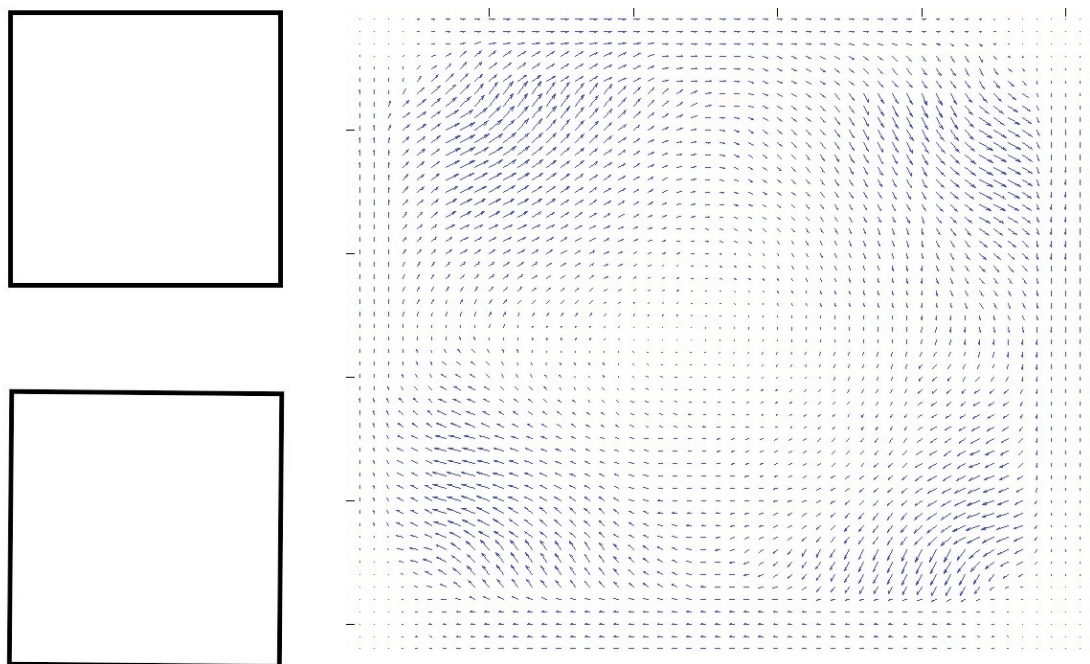
A fenti egyenlet átrendezésével kapjuk az alapegyenletet:

$$-f_t = \frac{\Delta x}{\Delta t} \cdot f_x + \frac{\Delta y}{\Delta t} \cdot f_y = (f_x, f_y) \cdot \left( \frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t} \right)^T = \nabla f \cdot c^T.$$

A cél a  $c = (u, v)$  sebesség-vektor meghatározása, mivel az  $f_t$  és a  $\nabla f = (f_x, f_y)$  gradiens kiszámítható, közelíthető  $f(x, y, t)$ -ből. A sebesség-vektor kiszámítására javasolt módszer teljes részletességgel megtalálható Horn és Schunck cikkében [33]. Itt csak annyit említünk, hogy a módszer az alábbi függvény minimalizálására fut ki:

$$E^2(x, y) = (u \cdot f_x + v \cdot f_y + f_t)^2 + \lambda \cdot (u_x^2 + u_y^2 + v_x^2 + v_y^2),$$

melyben az első tagban az alapegyenlet köszön vissza, a második pedig a simasági kritérium, amit a  $\lambda$  Lagrange-multiplikátor súlyoz.



9.4. ábra. Példa optikai áramlásra. Egy mozgókép két egymást követő kockája (bal) és a számított mozgás-mező (jobb).

A fenti egyenletből az alábbi parciális differenciál-egyenletek származnak:

$$\begin{aligned}(\lambda^2 + f_x^2) \cdot u + f_x \cdot f_y \cdot v &= \lambda^2 \cdot \bar{u} - f_x \cdot f_t \\(\lambda^2 + f_y^2) \cdot v + f_x \cdot f_y \cdot u &= \lambda^2 \cdot \bar{v} - f_y \cdot f_t,\end{aligned}$$

ahol  $\bar{u}$  és  $\bar{v}$  a sebesség átlaga az  $x$  és az  $y$  tengelyek irányában, az  $(x, y)$  pont valamely lokális környezetében.

A fenti parciális differenciál-egyenletek megoldását Gauss-Seidel iterációval már nem részletezzük.

Az optikai áramlás számítása hibás lesz, ha nem teljesülnek a konstans fényességre és a sebesség simaságára vonatkozó feltételezések. Sajnos a valós képeken a fenti feltételek gyakorta nem teljesülnek, így az optikai áramlás számítására globális relaxációs technikák alkalmazhatók.

Az optikai áramlással közelített mozgás-mezőket a 9.4. és a 9.5. ábrákkal illusztráljuk.

### 9.3. Mozgás-elemzés pont-megfeleltetéssel

Ha az érdekes, jellemző pontokat detektálni tudjuk a mozgókép egyes kockáin és azokat az egymást követő képpárokon meg tudjuk feleltetni egymásnak, úgy az érdekes pontok röppályáinak meghatározása, a mozgó objektumok követése már megoldottnak tekinthető. A mód-



9.5. ábra. Példa optikai áramlásra. Egy mozgókép két egymást követő kockája (bal) és a számított mozgás-mező (jobb).

szer kulcsa tehát a jellemző pontok robusztus detektálását megoldó automatikus algoritmusokon és a *megfeleltetési probléma* (*correspondence problem*) hatékony megoldása. Ilyen pl. az ICP (iterative closest point) algoritmus [6].

Jellemző pontoknak általában a *sarok-pontokat* (*corner point*) választják, melyek detektálására hatékony módszer pl. a Moravec sarok-detektor [44] és annak Harris és Stephens által javasolt javítása [27]. A megfeleltetési probléma megoldására javasolt módszerek [77] alkalmazásakor esetünkben kihasználhatjuk a következőket:

- Ha ismert az objektum maximális sebessége, akkor a objektum valamely pontjának helye a következő képkockán a korábbi pozíció köré írt adott sugarú körön belül marad.
- Ha ismert egy pont elmozdulásának iránya és sebessége az előző és az aktuális képkocka alapján, akkor a következő kockán is behatárolható a pozíciója, ha az irány- és a sebesség-változás (gyorsulás) mértéke ismert.
- Feltételezhetjük, hogy az objektum valamennyi pontja hasonlóképpen mozog, merev testek eltolásos mozgása esetén pedig egy tetszőleges pont elmozdulásának ismeretében az objektum valamennyi pontjának elmozdulását is ismerjük.

## 10. fejezet

# A többfelbontású feldolgozás és a wavelet-ek

Az 1950-es évek végét követő évtizedekben a transzformáció-alapú kép- és jelfeldolgozásban a Fourier-transzformációnak jutott vezető szerep. Napjainkra viszont a *wavelet* transzformációt tekintik olyan ígéretes eszköznek, mellyel könnyebben megoldható a képek analízise és tömörítése.

Míg a Fourier-transzformációnál a bázisfüggvények periodikusak (komplex szinuszoidok) és a transzformáció csak frekvencia-információt szolgáltat, addig a wavelet transzformáció változatos frekvenciájú és időtartamú „hullámocskák” (*wavelets*) alapul, így a wavelet transzformáció időbeli információt is hordoz.

A wavelet-ek a jel- és képfeldolgozás *többfelbontású* (*multiresolution*) módszereiben az 1980-as évek végén jelentek meg [40]. A jelen fejezetben is a wavelet-eknek a többfelbontású feldolgozásban betöltött szerepére koncentrálunk.

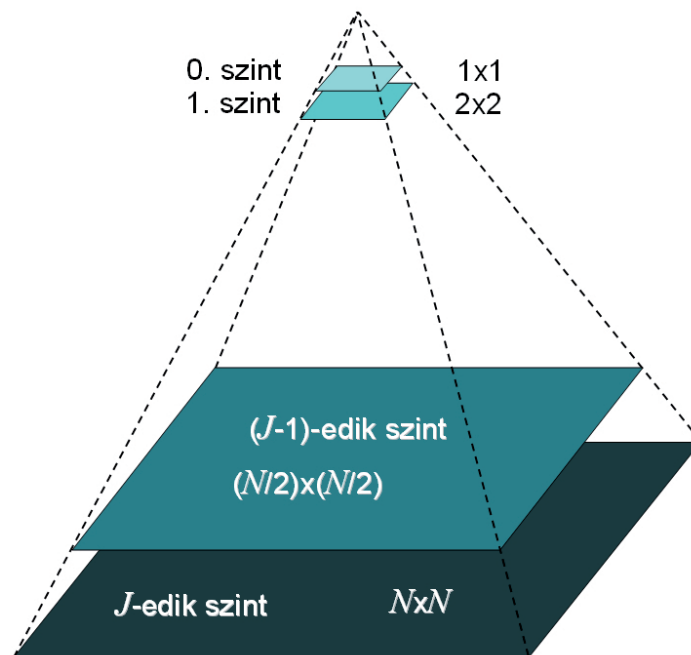
A terület feldolgozásakor alapvetően két mű, Gonzalez és Woods kézikönyve [24] valamint Watt és Policarpo műve [77] volt a forrásunk.

### 10.1. Kép-piramisok

Valamely kép több felbontásban (mintavételezési léptékben) történő ábrázolására hatékony és ugyanakkor egyszerű eszköz a *kép-piramis* (*image pyramid*) [15].

Egy kép-piramis valamely kép egyre csökkenő felbontású változatait tartalmazza piramisalakban elrendezve. A legalsó piramis-szinten, a piramis alapján maga az  $N \times N$ -es eredeti kép helyezkedik el, ahol  $N = 2^J$ . A piramis következő szintjén található kép mérete  $(N/2) \times (N/2)$ -es és minden további szintjén tovább feleződik a képméret (ld. 10.1. ábra). A piramis tehát  $J+1$  szintből áll, ahol  $J = \log_2 N$  és a  $p$ -edik szinten a képméret  $2^p \times 2^p$  ( $p=0, 1, \dots, J$ ).

A gyakorlatban a teljes piramist  $(P+1)$ -szintesre csonkolják (vagyis a felső  $J-P$  „kis” képet már figyelmen kívül hagyják). A kép-piramis valamely közbülső szintjén található kép csak negyedannyi pontból áll, mint az alatta lévő, így a csonkolt kép-piramis összes pontjára



10.1. ábra. Példa kép-piramisra.

teljesül az alábbi egyenlőtlenség:

$$N^2 \cdot \sum_{p=0}^P \frac{1}{4^p} \leq \frac{4}{3} \cdot N^2,$$

vagyis a teljes kép-piramis pontjainak száma a kiindulási képénél legfeljebb csak egyharmadnyival lehet több.

A kép-piramis felépítésekor valójában két piramist építünk: egyet, amiben maga a kép és annak egyre kisebb változatai szerepelnek, a másik pedig a *jóslat üledéke* (*prediction residual*) piramis. A kép-piramis előállítása értelemszerűen alulról-felfelé, a kezdeti képből kiindulva történik. A  $p$ -edik szinten lévő  $2^p \times 2^p$  méretű kép elkészülte után a következő, a  $(p-1)$ -edik szint ( $p = J, J-1, \dots, J-P+1$ ) képét határozzuk meg egy szűrést (pl. átlagolást, vagy aluláteresztő Gauss-szűrést) követő kicsinyítéssel (az  $(s_x, s_y) = (0.5, 0.5)$  faktorokkal történő skálázással). A kép-piramis ily módon előállított  $(p-1)$ -edik képét ezt követően „visszanagyítjuk” (az  $(s_x, s_y) = (2, 2)$  faktorokkal) egy alkalmasan megválasztott interpolációs technikával (ld. 7. fejezet), így megjósolva a  $p$ -edik szint képét (amit amúgy pontosan ismerünk). A  $p$ -edik szint tényleges képéből pontonként kivonva a jóslat-képet kapjuk a második, a jóslat üledéke piramis  $p$ -edik szintjének képét. A kép-piramist *Gauss-piramisnak* (*Gaussian pyramid*), a jóslat üledékét pedig *Laplace-piramisnak* (*Laplacian pyramid*) is nevezik. A piramis-építés menetét a 10.2. ábra illusztrálja.

A Gauss- és a Laplace-piramisokra a 10.3. és a 10.4. ábrákkal adunk példákat. Megjegyezzük, hogy a Laplace-piramis előjeles képekből áll, vagyis az intenzitás-értékek lehetnek negatív számok is. Ábráinkon a Laplace-piramis képein a „középszürke” felel meg a 0





10.2. ábra. Példa piramis-építésre. A Gauss-piramis  $p$ -edik szintjén lévő  $G_p$  kép (fent, bal), a Gauss-piramis szűrést követő kicsinyítéssel kapott  $G_{p-1}$  képe (fent, jobb), a  $G_{p-1}$  „visszanagyításával” kapott, a  $G_p$  képet megjósoló  $T_p$  kép (lent, bal) és a Laplace-piramis  $p$ -edik szintje, vagyis a  $G_p - T_p$  kép (lent, jobb).

intenzitás-értéknek. A „középszürkénél” sötétebb árnyalatú képpontok intenzitása negatív, a világosabbaké pedig pozitív.

A Gauss-piramis durvább felbontású szintjein a nagyméretű struktúrák, a többszörös simítást is „túlélő” jellemzők elemezhetők. Míg a magasabb szintek átfogó képet nyújtanak a kiindulási képről, addig a piramis aljának nagyfelbontású szintjein az egyedi objektumok jellemzői is kinyerhetők. A *durvától a finomig* (*coarse-to-fine*) analízis a képfeldolgozás és az alakfelismerés (mintaillesztés) számos alkalmazásánál hasznos. Pl. a Canny-féle éldetektor is alkalmazza a *jellemző-szintézis* (*feature synthesis*) technikát, kép-piramison dolgozik a hierarchikus chamfer-illesztés (ld. 5. fejezet) és az automatikus képregisztráció (ld. 8. fejezet) is gyorsítható azáltal, hogy a durvább szinteken (ahol a kisebb felbontású, a kiindulásnál jóval kevesebb pontból álló képeken) „rövid” idő alatt a keresési tér „számos” pontjára kiértékelhetjük a hasonlósági mértéket, majd a finomabb felbontású képeken már csak „kevés” eset vizsgálatával finomíthatjuk a durvábbakon kapott eredményeket.





10.3. ábra. Egy Gauss- (fent) és a hozzá tartozó Laplace-piramis (lent). A csonkolt piramisok 4-szintesek, vagyis  $P = 3$ .

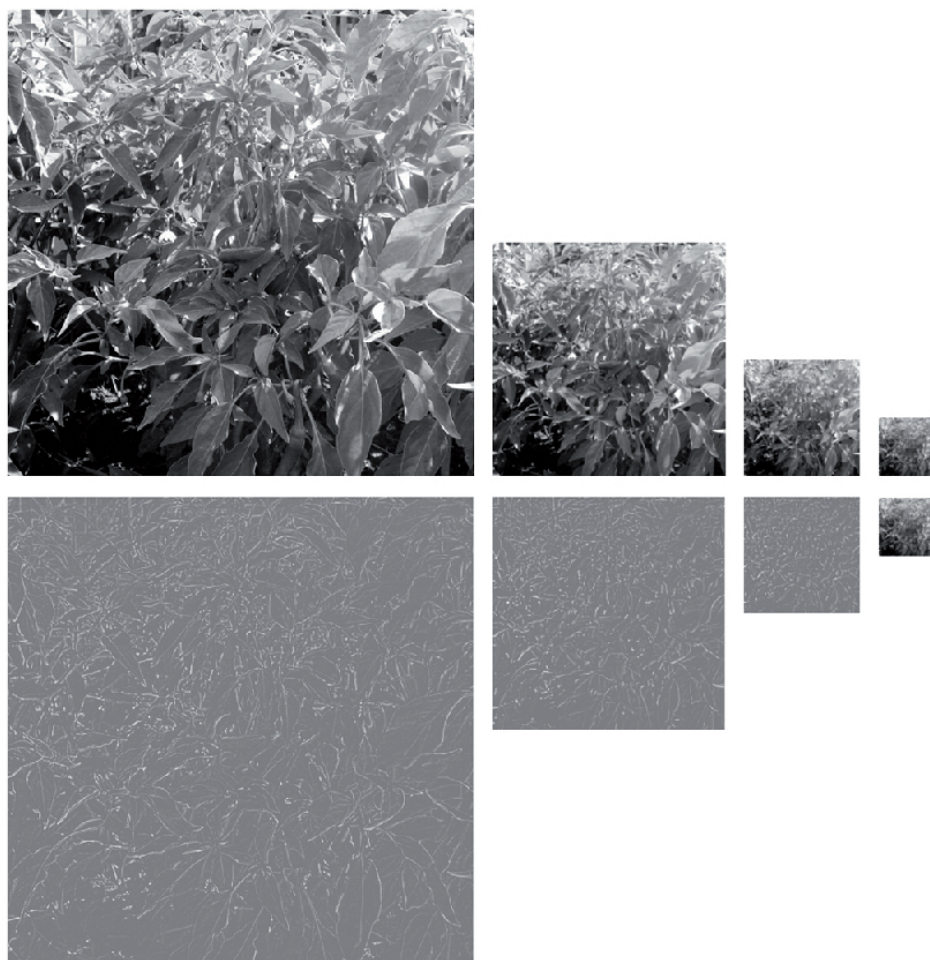
A Laplace-piramis előjeles képein (ld. a 10.2., a 10.3. és a 10.4. ábrákat) a predikciós hiba általában kicsi (a képek hisztogramjain a 0 intenzitásnál található a kiemelkedően magas csúcs), így a Laplace-piramis képei változó hosszúságú kódolással hatékonyan tömöríthetők.

## 10.2. A Haar transzformáció

A Haar transzformáció és a Haar bázis-függvények a többfelbontású feldolgozás mellett a wavelet transzformációban és a (vesztésmentes vagy veszteséges) képtömörítésben is fontos szerepet játszanak.

Az  $F$   $N \times N$ -es ( $N = 2^n$ ) kép Haar transzformációját a következő mátrix-egyenlet írja le:

$$T = H_N \cdot F \cdot H_N^T,$$



10.4. ábra. Egy Gauss- (fent) és a hozzá tartozó Laplace-piramis (lent). A csonkolt piramisok 4-szintesek, vagyis  $P = 3$ .

ahol (az ugyancsak  $N \times N$ -es)  $H_N$  transzformációs mátrix a Haar bázis-függvényekből épül fel:

$$H_N = \begin{bmatrix} h_0(0) & h_0(\frac{1}{N}) & \cdots & h_0(\frac{N-1}{N}) \\ h_1(0) & h_1(\frac{1}{N}) & \cdots & h_1(\frac{N-1}{N}) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1}(0) & h_{N-1}(\frac{1}{N}) & \cdots & h_{N-1}(\frac{N-1}{N}) \end{bmatrix}.$$

A  $H_N$  mátrix  $k$ -adik sorában szereplő  $h_k(z)$  ( $k = 0, 1, \dots, N-1$ ,  $z = 0, \frac{1}{N}, \dots, \frac{N-1}{N}$ ) függvények megadásához szükségünk van a  $p$  és a  $q$  változók bevezetésére, ahol:

- $k = 2^p + q - 1$ ,
- $p = 0, 1, \dots, N-1$ ,

- $q = 1, 2, \dots, 2^p$ , ha  $p \neq 0$  és
- $q = 0, 1$ , ha  $p = 0$ .

A két új változó segítségével definiált  $h_k(z) = h_{p,q}(z)$  Haar bázis-függvények:

$$h_0(z) = h_{0,0}(z) = \frac{1}{\sqrt{N}},$$

$$h_k(z) = h_{p,q}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2}, & \text{ha } (q-1)/2^p \leq z < (q-0.5)/2^p \\ -2^{p/2}, & \text{ha } (q-0.5)/2^p \leq z < q/2^p \\ 0, & \text{különben} \end{cases}$$

( $k = 1, \dots, N-1$ ,  $z = 0, \frac{1}{N}, \dots, \frac{N-1}{N}$ ).

Mielőtt példákat adnánk a  $H_N$  transzformációs mátrixokra, nézzük a  $k$  sorindexek ( $2^p + q - 1$ )-alakú felbontását:

$k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$p$	0	0	1	1	2	2	2	2	3	3	3	3	3	3	3	3
$q$	0	1	1	2	1	2	3	4	1	2	3	4	5	6	7	8

Könnyen ellenőrizhető, hogy a  $2 \times 2$ -es transzformációs mátrix a következő:

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

a  $4 \times 4$ -es esetben:

$$H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix},$$

a  $8 \times 8$ -as esetben pedig:

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}.$$

A kiindulási  $F$  kép a  $T$  Haar-transzformált kép ismeretében egyszerűen rekonstruálható, mivel a  $H_N$  transzformációs mátrix ortonormált, vagyis

$$H_N^{-1} = H_N^T.$$

Következésképpen:

$$F = H_N^T \cdot T \cdot H_N.$$

Tekintsük a legegyszerűbb  $N = 2$  esetben az

$$F = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

kép transzformáltját, vagyis a

$$\begin{aligned} T &= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b-c+d \end{bmatrix} \end{aligned}$$

mátrixot. Megállapíthatjuk, hogy a bal felső sarokban álló elem a sorok (vagy oszlopok) átlagának összege (vagyis tekinthetjük egy aluláteresztő- vagy egy átlag-szűrés eredményének), a jobb felső elem a vízszintes (vagy függőleges) irányú gradiensek átlaga, az alsó sor elemei pedig az átlós görbületek. (Megjegyzendő, hogy a  $H_2$  ( $n = 1$ ) mátrix szimmetrikus, de ez a tulajdonság már nem teljesül kettő magasabb hatványai ( $n > 2$ ) esetén a Haar-mátrixokra.)

Ha egy  $2^n \times 2^n$ -es képet  $2 \times 2$ -es blokkokra bontunk, majd minden egyes blokkot transzformálunk a  $H_2$  mátrixszal, akkor a transzformált blokkok azonos indexű elemeiből összeállíthatunk négy  $2^{n-1} \times 2^{n-1}$ -es képet (rendre a szűrt, a gradiens és az átlós görbületek képeit). A  $H_2$  transzformációs mátrix alkalmazásával tehát a kiindulási képből egyszerre négy kép-piramist is építhetünk.

### 10.3. Wavelet függvények

A wavelet függvények ismertetése előtt a függvények sorfejtésébe és a skálázó függvényekbe kell betekintenünk.

Az  $f(x)$  (egyváltozós) függvény vagy jel gyakran jobban elemezhető a következő alakban:

$$f(x) = \sum_k \alpha_k \cdot \varphi_k(x),$$

ahol  $k$  a véges vagy végtelen sorfejtés (*series expansion*) egész értékeket felvevő indexe, az  $\alpha_k$  valós számok a sorfejtés együtthatói, a  $\varphi_k(x)$  függvények pedig a sorfejtés függvényei. Ha a sorfejtés egyértelmű, vagyis valamely  $f(x)$  függvényhez csak egyetlen  $\{\alpha_k\}_k$  együtthatóhalmaz tartozik, akkor a  $\varphi_k(x)$  függvényeket bázis-függvényeknek nevezzük. A  $\{\varphi_k(x)\}_k$  függvényhalmazból lineáris kombinációval előálló függvények a  $V$  függvény-osztályhoz tartoznak. Esetünkben tehát  $f(x) \in V$ .

A sorfejtés  $\alpha_k$  együtthatói a  $\{\varphi_k(x)\}_k$  bázis-függvényekhez tartozó  $\{\tilde{\varphi}_k(x)\}_k$  duális függvényekkel határozhatók meg:

$$\alpha_k = \langle \tilde{\varphi}_k(x), f(x) \rangle = \int \tilde{\varphi}_k^*(x) \cdot f(x) dx,$$

ahol a  $\tilde{\varphi}_k^*(x)$  függvény a  $\varphi_k(x)$  komplex konjugáltja.

Tekintsük a  $\varphi_{j,k}(x)$  függvényeket, melyek a négyzetesen integrálható  $\varphi(x)$  valós függvényből bináris skálázással és eltolással kapunk:

$$\varphi_{j,k}(x) = 2^{j/2} \cdot \varphi(2^j x - k),$$

ahol  $j$  és  $k$  egészek, melyek közül  $k$  határozza meg  $\varphi_{j,k}(x)$  pozícióját az  $x$  tengely mentén,  $j$  adja meg a függvény szélességét (ugyancsak az  $x$  tengely mentén) és  $2^{j/2}$  vezérli a függvény magasságát vagy amplitúdóját.

Helyettesítsük be rendre a  $j$  értékeinek a 0, a  $\pm 1$  és a  $\pm 2$  egészeket. A kapott függvények:

$$\begin{aligned}\varphi_{-2,k}(x) &= \varphi(x/2 - k)/\sqrt{2}, \\ \varphi_{-1,k}(x) &= \varphi(x/4 - k)/2, \\ \varphi_{0,k}(x) &= \varphi(x - k), \text{ ennél fogva } \varphi_{0,0}(x) = \varphi(x), \\ \varphi_{1,k}(x) &= \sqrt{2}\varphi(2x - k), \\ \varphi_{2,k}(x) &= 2\varphi(4x - k).\end{aligned}$$

Mivel a  $\varphi_{j,k}(x)$  függvény alakja  $j$ -től függ, így a  $\varphi(x)$  függvényt *skálázó függvénynek* (*scaling function*) nevezzük. Egy rögzített  $j = j_0$ -ra a  $\{\varphi_{j_0,k}(x)\}_k$  függvényekből lineáris kombinációval előállítható függvényeket a  $V_{j_0}$  függvény-osztályhoz soroljuk.

A továbbiakban skálázó függvényként vizsgáljuk az egységnyi szélességű és egységnyi magasságú *Haar skálázó függvényt*:

$$\varphi(x) = \begin{cases} 1, & \text{ha } 0 \leq x < 1 \\ 0, & \text{különben} \end{cases}.$$

A *többsfelbontású vizsgálat* (*multiresolution analysis*) a skálázó függvényekkel szemben támasztott követelményei közül az első az, hogy legyenek ortogonálisak az egész léptékű eltolásra. A Haar skálázó függvény teljesíti ezt az elvárást, mivel

$$\varphi(x - k_1) \cdot \varphi(x - k_2) = 0$$

tetszőleges  $k_1 \neq k_2$  (egész) eltolási tényezők esetén.

A többsfelbontású vizsgálat további elvárása az, hogy a  $\{\varphi_{j,k}(x)\}_k$  függvényekből előállítható  $V_j$  függvény-osztályokra teljesüljön a következő tartalmazkodás:

$$V_{-\infty} \subset \dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset V_{\infty}.$$

A  $\varphi(x)$  Haar skálázó függvényből felépített  $\{\varphi_{0,k}(x)\}_k$  függvények előállíthatók a  $\{\varphi_{1,k}(x)\}_k$  függvények alábbi lineáris kombinációjaként:

$$\varphi_{0,k}(x) = \frac{1}{\sqrt{2}}\varphi_{1,2k}(x) + \frac{1}{\sqrt{2}}\varphi_{1,2k+1}(x).$$

Következésképpen valamennyi  $f(x) \in V_0$  függvényre  $f(x) \in V_1$ , azaz  $V_0 \subset V_1$ .

Az is belátható, hogy valamennyi  $f(x)$  függvényre és valamennyi  $j$ -re, ha  $f(x) \in V_j$ , akkor  $f(2x) \in V_{j+1}$ , vagyis a Haar skálázó függvényre teljesül a többfelbontású vizsgálat második kritériuma is.

Mivel  $\varphi_{j,k}(x) \in V_j \subset V_{j+1}$ , így

$$\varphi_{j,k}(x) = \sum_n \alpha_n \cdot \varphi_{j+1,n}(x) = \sum_n h_\varphi(n) \cdot 2^{(j+1)/2} \cdot \varphi(2^{j+1}x - n),$$

ahol először a  $k$  indexet cseréltük  $n$ -re, majd az  $\alpha_n$  együtthatót  $h_\varphi(n)$ -re.

Mivel  $\varphi(x) = \varphi_{0,0k}(x)$ , így a fenti egyenletbe a  $\varphi_{0,0k}(x)$  függvényt helyettesítve láthatjuk azt, hogy maga a  $\varphi(x)$  Haar skálázó függvény is sorba-fejthető:

$$\varphi(x) = \sum_n h_\varphi(n) \sqrt{2} \varphi(2x - n),$$

így a  $h_\varphi(n)$  együtthatókból álló  $h_\varphi$  vektort *skálázó vektornak (scaling vector)* nevezik.

A  $h_\varphi$  skálázó vektor első két eleme a  $2 \times 2$ -es  $H_2$  mátrix első sorával egyezik meg, azaz:

$$[h_\varphi(0), h_\varphi(1)] = [h_0(0), h_0(1/2)] = [1/\sqrt{2}, 1/\sqrt{2}].$$

Így a  $\varphi(x)$  Haar skálázó függvény a következő alakban írható fel:

$$\varphi(x) = \frac{1}{\sqrt{2}} \sqrt{2} \varphi(2x) + \frac{1}{\sqrt{2}} \sqrt{2} \varphi(2x - 1) = \varphi(2x) + \varphi(2x - 1).$$

Ezek után készen állunk arra, hogy definiálhassuk a  $\psi(x)$  *wavelet függvényt*, mely a  $\{\psi_{k,j}(x)\}$  *waveletekkel* a  $W_j = V_{j+1} - V_j$  és a  $V_j$  függvényosztály tagjainak sorfejtését oldja meg.

A  $\psi(x)$  wavelet függvényből bináris skálázással és eltolással kapott  $\psi_{k,j}(x)$  *wavelet* az alábbi formulával számítható:

$$\psi_{j,k}(x) = 2^{j/2} \cdot \psi(2^j x - k),$$

az  $f(x) \in W_j$  függvény sorfejtése pedig:

$$f(x) = \sum_k \alpha_k \cdot \psi_{j,k}(x).$$

Magának a  $\psi(x)$  wavelet függvénynek a sorfejtése a  $\varphi(x)$  skálázó függvénnyel:

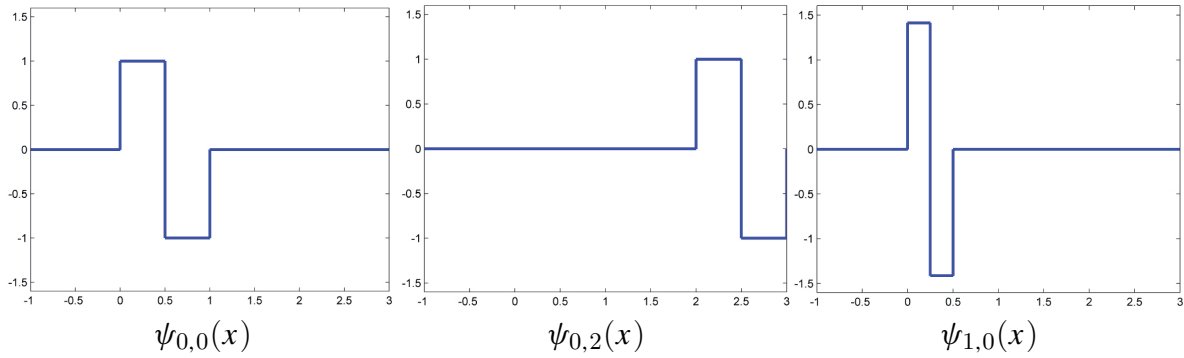
$$\psi(x) = \sum_n h_\psi(n) \sqrt{2} \varphi(2x - n),$$

ahol a  $h_\psi(n)$  *wavelet függvény együtthatókból* áll a  $h_\psi$  *wavelet vektor*.

A wavelet vektor kifejezhető a skálázó vektorral:

$$h_\psi(n) = (-1)^n h_\varphi(1 - n).$$





10.5. ábra. Példák Haar wavelet függvényekre.

A wavelet vektor első két eleme a  $h_\varphi$  skálázó vektor korábban megadott első két eleme alapján:

$$[h_\psi(0), h_\psi(1)] = [1/\sqrt{2}, -1/\sqrt{2}].$$

A Haar wavelet függvény tehát

$$\psi(x) = \varphi(2x) - \varphi(2x - 1),$$

azaz

$$\psi(x) = \begin{cases} 1, & \text{ha } 0 \leq x < 0.5 \\ -1, & \text{ha } 0.5 \leq x < 1. \\ 0, & \text{különben} \end{cases}$$

Haar waveletekre a 10.5. ábra mutat három példát.

Könnyen belátható, hogy a Haar transzformációs mátrixok sorai megadhatók a Haar skálázó függvénnyel, a Haar wavelet függvénnyel és a Haar waveletekkel. Nézzük például a  $8 \times 8$ -as mátrixot:

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

$$= \begin{bmatrix} \varphi(0) & \varphi(1/8) & \dots & \varphi(7/8) \\ \psi(0) & \psi(1/8) & \dots & \psi(7/8) \\ \psi_{1,0}(0) & \psi_{1,0}(1/8) & \dots & \psi_{1,0}(7/8) \\ \psi_{1,1}(0) & \psi_{1,1}(1/8) & \dots & \psi_{1,1}(7/8) \\ \psi_{2,0}(0) & \psi_{2,0}(1/8) & \dots & \psi_{2,0}(7/8) \\ \psi_{2,1}(0) & \psi_{2,1}(1/8) & \dots & \psi_{2,1}(7/8) \\ \psi_{2,2}(0) & \psi_{2,2}(1/8) & \dots & \psi_{2,2}(7/8) \\ \psi_{2,3}(0) & \psi_{2,3}(1/8) & \dots & \psi_{2,3}(7/8) \end{bmatrix}.$$

## 10.4. Wavelet transzformáció

A *diszkrét wavelet transzformáció* (*discrete wavelet transform (DWT)*) ismertetése előtt be kell vezetni a négyzetesen integrálható  $f(x)$  függvény *wavelet sorfejtését* (*wavelet series expansion*) a  $\psi(x)$  wavelet függvénnyel és a  $\varphi(x)$  skálázó függvénnyel. A sorfejtést a következő formula írja le:

$$f(x) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j,k}(x),$$

ahol a  $c_{j_0}(k)$  súlyokat a *közelítésnek* (*approximation*) vagy *skálázó együtthatóknak* (*scaling coefficient*) nevezzük, a  $d_j(k)$  súlyok pedig a *wavelet együtthatók* (*wavelet coefficient*). Az együtthatók az alábbi egyenletek szerint számíthatók:

$$c_{j_0}(k) = \langle f(x), \tilde{\varphi}_{j_0,k}^*(x) \rangle = \int f(x) \tilde{\varphi}_{j_0,k}^*(x) dx,$$

$$d_j(k) = \langle f(x), \tilde{\psi}_{j,k}^*(x) \rangle = \int f(x) \tilde{\psi}_{j,k}^*(x) dx,$$

ahol a  $\tilde{\varphi}_{j_0,k}^*(x)$  és a  $\tilde{\psi}_{j,k}^*(x)$  duális függvények rendre a  $\varphi_{j_0,k}(x)$  és a  $\psi_{j,k}(x)$  komplex konjugáltjai.

A sorfejtés első tagja az  $f(x)$  függvénynek a  $j_0$  skála szerint adja meg egy közelítést (vagy pontosan az  $f(x)$  függvényt, amennyiben  $f(x) \in V_{j_0}$ ), a második tag pedig a finomabb skálájú wavelet függvényekkel növeli a közelítés pontosságát.

Az  $M$  alappontos (1-dimenziós) DWT – hasonlóan a diszkrét Fourier transzformációhoz – az  $f(x)$ , a  $\varphi_{j_0,k}(x)$  és a  $\psi_{j,k}(x)$  függvények az  $x$  diszkrét változó értékeit a  $\{0, 1, \dots, M-1\}$  halmazból veszi, ahol általában a  $j_0 = 0$  és az  $M = 2^J$ .

A DWT transzformáció formulái:

$$W_{\varphi}(j_0, k) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} f(x) \varphi_{j_0,k}(x),$$

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} f(x) \psi_{j,k}(x),$$

ahol  $j \geq j_0$ ,  $W_{\varphi}(j_0, k)$  és  $W_{\psi}(j, k)$  pedig rendre megfelelnek a wavelet sorfejtés  $c_{j_0}(k)$  és a  $d_j(k)$  együtthatóinak. (Az integrálok helyébe a szummák és az  $1/\sqrt{M}$  normalizáló tényezők kerültek.)

A  $W_{\varphi}(j_0, k)$  és  $W_{\psi}(j, k)$  együtthatókból az *inverz DWT transzformációval* az  $f(x)$  függvényt kapjuk vissza:

$$f(x) = \frac{1}{\sqrt{M}} \sum_k W_{\varphi}(j_0, k) \varphi_{j_0,k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_{\psi}(j, k) \psi_{j,k}(x).$$

A fenti formulák csak ortonormált bázisfüggvények esetén érvényesek, más esetekben a  $\varphi_{j_0,k}(x)$  és a  $\psi_{j,k}(x)$  függvényeket a komplex konjugáltjaikkal kell helyettesíteni.

Haar wavelet-ek esetén a diszkrét skálák és wavelet függvények megfelelnek az  $M \times M$ -es Haar mátrix sorainak. Ekkor a transzformációnál a minimális skála 0 ( $j_0 = 0$ ), a maximális pedig  $J - 1$  ( $J = \log_2 M$ ,  $M = 2^J$ ). Megemlítendő, hogy – csakúgy, mint a diszkrét Fourier transzformációra – a DWT-re is terveztek gyors algoritmust (*fast wavelet transform (FWT)*) [41]. Az FWT numerikus komplexitása  $M$ -elemű diszkrét vektorra  $\mathcal{O}(M)$  – szemben a konkurrens gyors Fourier transzformáció, az FFT  $\mathcal{O}(M \log_2 M)$  időigényével.

Az 1-dimenziós DWT-t – hasonlóan, mint a Fourier transzformációt – kiterjesztették 2-dimenziós függvényekre, képekre. A 2-dimenziós DWT az alábbi négy (2-változós) függvényt alkalmazza:

- **skálázó függvény:**

$$\varphi(x, y) = \varphi(x) \cdot \varphi(y),$$

- **vízszintes irányú változásokra érzékeny wavelet:**

$$\psi^H(x, y) = \psi(x) \cdot \varphi(y),$$

- **függőleges irányú változásokra érzékeny wavelet:**

$$\psi^V(x, y) = \varphi(x) \cdot \psi(y),$$

- **átlós irányú változásokra érzékeny wavelet:**

$$\psi^D(x, y) = \psi(x) \cdot \psi(y).$$

Az 1-dimenziós skálázó és wavelet függvények kiterjesztése az alábbi skálázott és eltoló bázis-függvényeket adja:

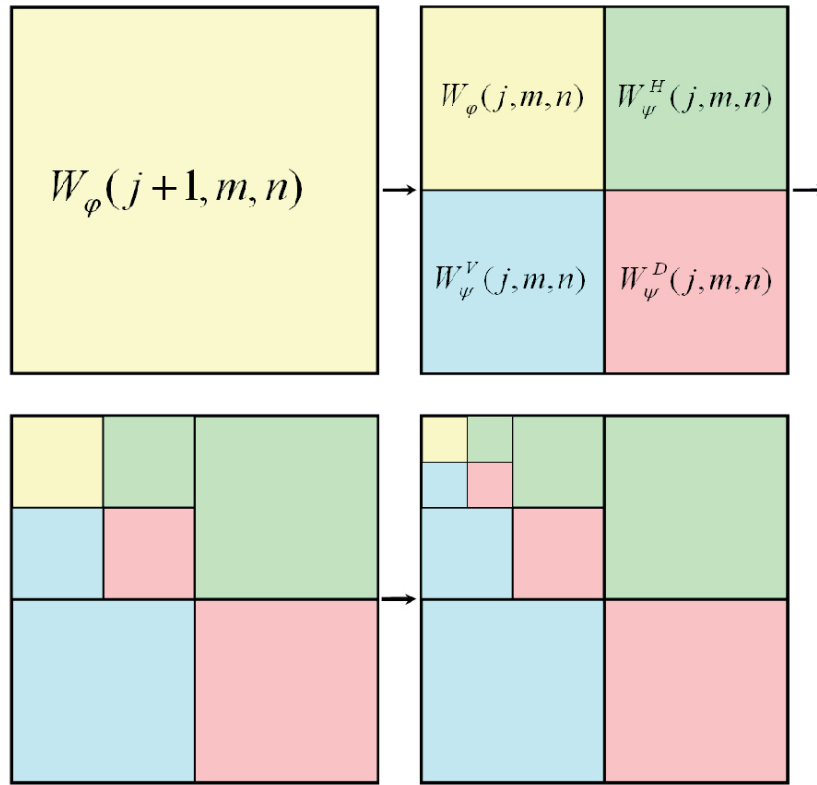
$$\begin{aligned}\varphi_{j,m,n}(x, y) &= 2^{j/2} \varphi(2^j x - m, 2^j y - n), \\ \psi_{j,m,n}^i(x, y) &= 2^{j/2} \psi^i(2^j x - m, 2^j y - n),\end{aligned}$$

ahol az  $i$  felső index a wavelet-ek irányát adja meg:  $i \in \{H, V, D\}$ .

Az  $M \times N$ -es  $f(x, y)$  képre a DWT az alábbi formulákkal adott:

$$\begin{aligned}W_\varphi(j_0, m, n) &= \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0,m,n}(x, y), \\ W_\psi^i(j, m, n) &= \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j,m,n}^i(x, y) \quad i \in \{H, V, D\}.\end{aligned}$$

Az 1-dimenziós esethez hasonlóan, a  $W_\varphi(j_0, m, n)$  együtthatók  $f(x, y)$  egy közelítését adják a  $j_0$  skálán, a  $W_\psi^i(j, m, n)$  együtthatók pedig vízszintes, függőleges és átlós részleteket adnak ahhoz a  $j \geq j_0$  skálákon. Általában  $j_0 = 0$  a kezdő skála, a maximális pedig  $J - 1$  és  $M = N = 2^J$ .



10.6. ábra. Képek felbontása DWT/FWT-vel három iteráción keresztül.

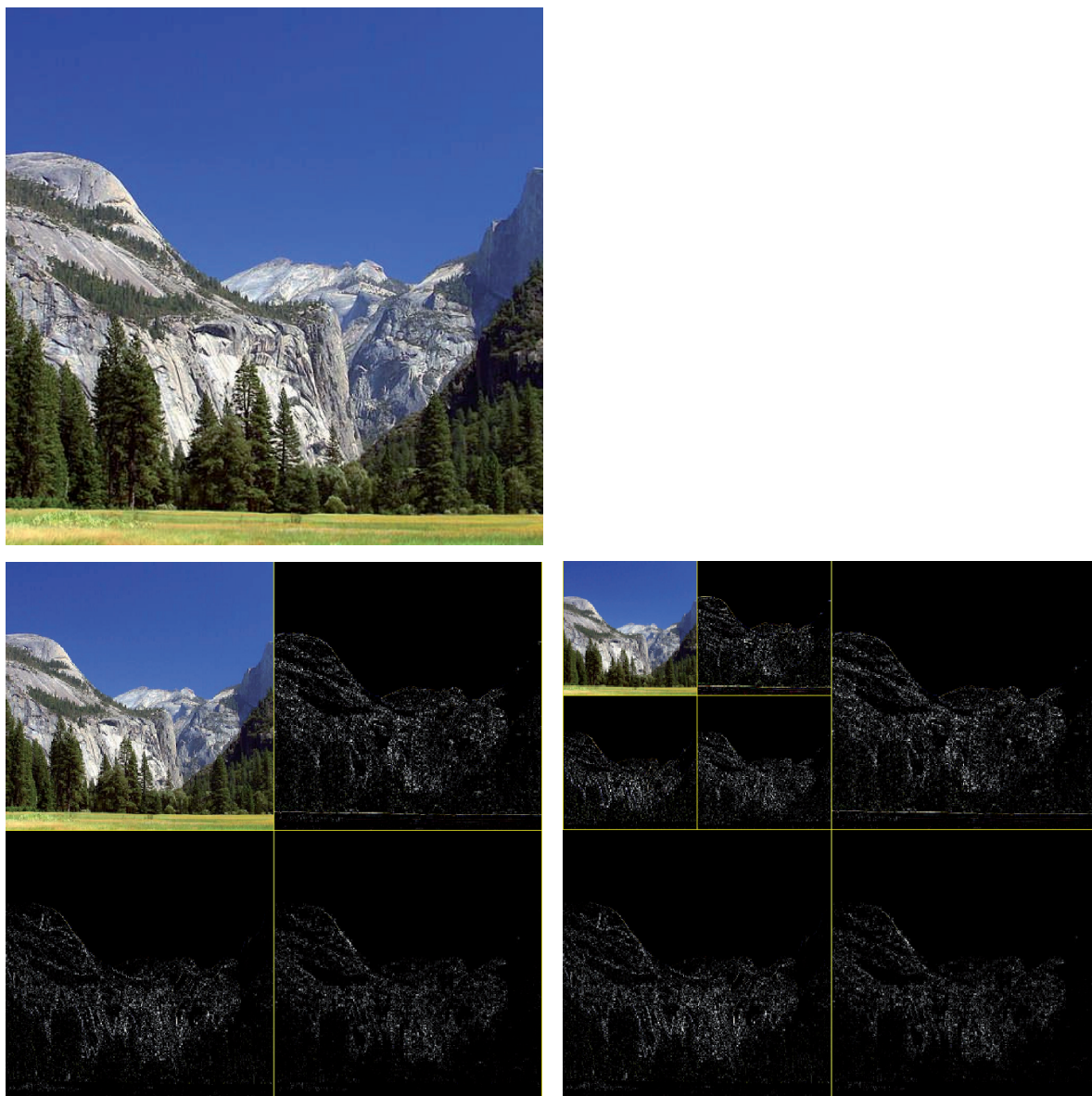
A 2-dimenziós inverz DWT az  $f(x, y)$  képet adja vissza a skálázó és wavelet együtthatók ismeretében:

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n) \varphi_{j_0, m, n}(x, y) + \frac{1}{\sqrt{MN}} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n) \psi_{j, m, n}^i(x, y).$$

Mivel a 2-dimenziós DWT szeparálható skálázó függvényt és (irány-érzékeny) wavelet-eket használ, így egy 2-dimenziós képre a DWT megoldható két lépésben: először a sorokra hajtsunk végre 1-dimenziós transzformációkat, majd a transzformált sorokból álló kép oszlopaira ugyancsak 1-dimenziós transzformációkat. Amennyiben az 1-dimenziós DWT gyors változatát, az FWT-t alkalmazzuk, úgy a 2-dimenziós FWT-hez jutunk.

A 2-dimenziós DWT a  $2^J \times 2^J$  méretű képből négy darab  $2^{J-1} \times 2^{J-1}$ -es képet állít elő. Amennyiben  $P$ -szer iterálunk, vagyis rendre a  $j=J-1, J-2, \dots, J-P$  skálákkal dolgozunk, úgy négy kép-piramishoz jutunk. Ha  $f(x, y) = W_\varphi(J, m, n)$ , úgy azt a 2-dimenziós DWT/FWT az egyenként negyedannyi pontból álló  $W_\varphi(J-1, m, n)$ ,  $W_\psi^H(J-1, m, n)$ ,  $W_\psi^V(J-1, m, n)$  és  $W_\psi^D(J-1, m, n)$  képekre bontja fel, melyek szokásos elrendezését (egyetlen  $2^J \times 2^J$  méretű képpé) a 10.6. ábrán mutatjuk be.

A kép-piramisok vagy a felbontott képek elemeinél a  $W_\varphi$  képeket méretcsökkentő aluláteresztő szűréssel kapjuk, míg a  $W_\psi^H(J-1, m, n)$ , a  $W_\psi^V(J-1, m, n)$  és a  $W_\psi^D(J-1, m, n)$



10.7. ábra. Egy kép (fent) és DWT/FWT-vel történő felbontásra két iterációval (lent).

képeket méretcsökkentő felüláteresztő szűrések eredményei. Ezeken a képeken (együtthatókon) végrehajtott alkalmas manipulációkat követő inverz transzformációval a kiindulási képen számos képfeldolgozási problémára (pl. éldetektálás, zajszűrés) is megoldást kapunk. A DWT legismertebb alkalmazása a képtömörítés egy új, a böngészők által ma még nem támogatott szabványa a JPEG 2000 (ld. <http://www.jpeg.org/jpeg2000/>).

A jól ismert „hagyományos” JPEG képtömörítés a képet (egymást nem átfedő) blokkokra osztja, majd az egyes blokkok diszkrét koszinusz transzformációval (DCT) kapott együtthatóit kvantálja (ekkor jöhet be a veszteség), majd a kvantált együtthatókat (veszteségmentes) entrópia-kódolásnak veti alá. A JPEG 2000 nem csak „jobban” tömöríthetők a képek észlelhető minőségromlás nélkül, hanem rugalmasabb is, mivel a képet nem rögzített méretű blokkokra, hanem változó méretű csempékre bonthatja (akár az egész kép is lehet egyetlen

csempe), majd az egyes csempékből DWT-vel kapott együtthatókat kvantálja. A JPEG 2000 részletes ismertetésétől – sajnos – most el kell tekintenünk.

Végezetül egy kép DWT/FWT-vel történő felbontását a 10.7. ábrával mutatjuk be.





# Irodalomjegyzék

- [1] Arun, K.S., Huang, T.S., Blostein, S.D.: Least-Squares Fitting of Two 3-D Point Sets, *IEEE Trans. Pattern Analysis and Machine Intelligence* **9** (1987) 698–703.
- [2] Attali, D., Montanvert, A.: Semicontinuous skeletons in 2D and 3D shapes, in *Proc. 2nd Int. Workshop on Visual Form*, World Scientific, 1994, 32–41.
- [3] Aurenhammer, F.: Voronoi diagrams: A survey of a fundamental geometric data structure, *ACM Comput. Surveys* **23** (1985) 345–405.
- [4] Balázs, P.: Képrekonstrukció, elektronikus jegyzet, 2011.
- [5] Baxes, G.A.: Digital Image Processing: Principles and Applications, John Wiley & Sons, 1994.
- [6] Besl, P.J., McKay, H.D.: A method for registration of 3-D shapes, *IEEE Trans. Pattern Analysis and Machine Intelligence* **14** (1992) 239–256.
- [7] Blum, H.: A transformation for extracting new descriptors of shape, Symposium on Models for the perception of Speech and Visual Form, 1964.
- [8] Blum, H.: Biological shape and visual science, *J. Theoretical Biology* **38** (1967) 205–287.
- [9] Bookstein, F.L.: Principal Warps: Thin-Plate Splines and the Decomposition of Deformations *IEEE Trans. Pattern Analysis and Machine Intelligence* **11** (1989) 567–585.
- [10] Borgefors, G.: Distance transformations in arbitrary dimensions, *Computer Vision, Graphics, and Image Processing* **27** (1984) 321–345.
- [11] Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm, *IEEE Trans. Pattern Analysis and Machine Intelligence* **10** (1988) 849–865.
- [12] Bow, S.-T.: Pattern recognition and image preprocessing, Marcel Dekker, Inc., 1992.
- [13] Brandt, J.W., Algazi, V.R.: Continuous skeleton computation by Voronoi diagram, *CV-GIP: Image Understanding* **55** (1992) 329–338.
- [14] Brown, L.G.: A survey of image registration techniques, *ACM Computing Surveys* **24** (1992) 325–376.

- [15] Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code, *IEEE Trans. Pattern Analysis and Machine Intelligence* **31** (1983) 532–540.
- [16] Calabi, L.: A study of the skeleton of plane figures, Technical report 60429, Parke Mathematical Laboratories, 1965.
- [17] Ciesielski, K.C., Udupa, J.K., Saha, P.K., Zhuge, Y.: Iterative relative fuzzy connectedness for multiple objects with multiple seeds, *Computer Vision and Image Understanding* **107** (2007) 160–182.
- [18] Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., Marchal, G.: Automated multi-modality image registration based on information theory, in *Proc. Information Processing in Medical Imaging*, 1995, 263–274.
- [19] Crane, R.: A simplified approach to image processing: Classical and modern techniques in C, Prentice Hall PTR, 1997.
- [20] Cuisenaire, O., Macq, B.: Fast Euclidean distance transformation by propagation using multiple neighborhoods, *Computer Vision and Image Understanding* **76** (1999) 163–172.
- [21] Danielsson, P.E.: Euclidean distance mapping, *CVGIP: Graphical Models and Image Processing* **14** (1980) 227–248.
- [22] Dougherty, E.R., Lotufo, R.A.: Hands-on Morphological Image Processing, SPIE PRESS Vol. TT59, 2003.
- [23] Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics – Principles and Practice, Addison-Wesley Publishing Company, Reading, Massachusetts, 1991.
- [24] Gonzalez, R.C., Woods, R.E.: Digital Image Processing (3rd Edition), Prentice Hall, 2007.
- [25] Goutsias, J., Heijmans, H.J.A.M. (eds.): Mathematical Morphology, IOS Press, 2000.
- [26] Hall, R.W.: Parallel connectivity-preserving thinning algorithms. in: Topological algorithms for digital image processing, Kong, T.Y., Rosenfeld, A. (Eds.), Elsevier Science (1996) 145–179.
- [27] Harris, C., Stephens, M.: A combined corner and edge detector, in *Proc. 4th Alvey Vision Conference*, 1988, 147–151.
- [28] Herman, G.T.: Geometry of Digital Spaces, Birkhäuser, Boston, 1998.
- [29] Herman, G.T., De Carvalho, B.M.: Multiseeded segmentation using fuzzy connectedness, *IEEE Trans. Pattern Analysis and Machine Intelligence* **23** (2001) 460–474.
- [30] Herman, G.T., Kuba, A. (eds.): Discrete Tomography: Foundations, Algorithms and Applications, Birkhäuser, Boston, 1999.

- [31] Herman, G.T., Kuba, A. (eds.): *Advances in Discrete Tomography and its Applications*, Birkhäuser, Boston, 2007.
- [32] Hill, D.L.G., Batchelor, P.G., Holden, M., Hawkes, D.J.: Medical image registration, *Physics in Medicine and Biology* **46** (2001) R1–R45.
- [33] Horn, B.K.P. Schunck, B.G.: Determining optical flow, *Artificial Intelligence* **17** (1981) 185–204.
- [34] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models, *Int. Journal of Computer Vision* **1** (1988) 321–331.
- [35] Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey, *Computer Vision, Graphics, and Image Processing* **48** (1989) 357–393.
- [36] Lam, L., Lee, S.-W., Suen, C.Y.: Thinning methodologies – A comprehensive survey, *IEEE Trans. Pattern Analysis and Machine Intelligence* **14** (1992) 869–885.
- [37] Lavallée, S.: Registration for computer-integrated surgery: methodology, state of the art, Taylor, R. H., Lavallée, S., Burdea, G.C., Mösges, R. (eds.), *Computer-integrated surgery, Technology and clinical applications*, MIT Press, Cambridge, MA, 1996, 77–97.
- [38] Mahling, D.H.: Coordinate systems and map projections for GIS, Maguire, D., Goodchild, M.F., Rhind, D.W. (eds.), *Geographical information systems – Principles and Applications*, Longman Scientific and Technical, 1991.
- [39] Maintz, J.B.A., Viergever, M.A.: A survey of medical image registration, *Medical Image Analysis* **2** (1998) 1–36.
- [40] Mallat, S.G.: A Compact Multiresolution representation: The wavelet model, in *IEEE Proceedings of Workshop on Computer Vision*, 1987, 2–7.
- [41] Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Trans. Pattern Analysis and Machine Intelligence* **11** (1989) 674–693.
- [42] Matheron, G.: *Random sets and integral geometry*, John Wiley & Sons, 1975.
- [43] Maurer, C.R., M. Fitzpatrick, J.: A review of medical image registration, Maciunas, R.J. (ed.), *Interactive image-guided neurosurgery*, American Association of Neurological Surgeons, 1993.
- [44] Moravec, H.: *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, Technical Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute, 1980.
- [45] Näf, M., Székely, G., Kikinis, R., Shenton, M.E., Kübler, G.: 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape, *Computer Vision, Graphics, and Image Processing* **66** (1997) 147–161.

- [46] Osher, S., Sethian, J.A. : Fronts propagating with curvature dependent speed : Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics* **79** (1988) 12–49.
- [47] Palágyi, K. : Vékonyító algoritmusok 3D képekre, Doktori értekezés, SZTE, Szeged, 2000.
- [48] Palágyi, K., Tschirren, J., Hoffman, E.A., Sonka, M. : Quantitative analysis of pulmonary airway tree structures, *Computers in Biology and Medicine* **36** (2006) 974–996.
- [49] Preparata, F.P., Shamos, M.I. : Computational Geometry, Springer-Verlag, 1990.
- [50] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. : Numerical Recipes in C: The Art of Scientific Computing (2nd edition), Cambridge University Press, 1992.
- [51] Roerdink, J.B.T.M., Meijster, A. : The watershed transform : Definitions, algorithms and parallelization strategies, *Fundamenta Informaticae* **41** (2001) 187–228.
- [52] Rosenfeld, A. : A characterization of parallel thinning algorithms, *Information and Control* **29** (1975) 286–291.
- [53] Russ, J.C. : The Image Processing Handbook, Fourth Edition, CRC Press, 2002.
- [54] Ryser, H.J. : Combinatorial properties of matrices of zeros and ones, *Canad. J. Math.* **9** (1957) 371–377.
- [55] Saito, T., Toriwaki, J. : A sequential thinning algorithm for three dimensional digital pictures using the Euclidean distance transformation, in *Proc. 9th Scandinavian Conf. Image Analysis, SCIA'95*, 1995, 507–516.
- [56] Serra, J. : Image analysis and mathematical morpholog, Academic Press, 1982.
- [57] Sethian, J.A. : A fast marching level set method for monotonically advancing fronts, in *Proc. National Academy of Sciences of the United States of America* 93/4 (1996) 1591–1595.
- [58] Sethian, J.A. : Level Set Methods – Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, 1996.
- [59] Shaked, D., Bruckstein, A.M. : Pruning medial axes, *Computer Vision and Image Understanding* **69** (1998) 156–169.
- [60] Shapiro, L. : Connected component labeling and adjacency graph, in *Topological algorithms for digital image processing*, T.Y. Kong and A. Rosenfeld (Editors), Elsevier, 1996.
- [61] Sharma G. (ed.) : Digital Color Imaging Handbook, CRC Press, 2002.
- [62] Siddiqi, K., Pizer, S. (eds.) : Medial representations – Mathematics, algorithms and applications, *Computational Imaging and Vision*, Vol. 37, Springer, 2008.

- [63] Soille, P.: *Morphological Image Analysis: Principles and Applications* (2nd edition), Springer, 2002.
- [64] Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis, and Machine Vision* (3rd edition), CL-Engineering, 2007.
- [65] Studholme, C., Hill, D.L.G., Hawkes, D.J.: An Overlap Invariant Entropy Measure of 3D Medical Image Alignment, *Pattern Recognition* **32** (1999) 71–86.
- [66] Suen, C.Y., Wang, P.S.P. (eds.): *Thinning methodologies for pattern recognition*, Series in Machine Perception and Artificial Intelligence **8**, World Scientific, 1994.
- [67] Svensson, S., Nyström, I., Borgefors, G.: Fully reversible skeletonization for volume images based on anchor-points from the  $D^{26}$  distance transform, in *Proc. 11th Scandinavian Conf. Image Analysis, SCIA'99*, 1999, 601–608.
- [68] Tanács, A., Czédli, G., Palágyi, K., Kuba, A.: Affine matching of two sets of points in arbitrary dimensions, *Acta Cybernetica* **15** (2001) 101–106.
- [69] Tanács, A.: Regisztrációs módszerek és orvosi alkalmazásaik, Doktori értekezés, SZTE, Szeged, 2007.
- [70] Tsao, Y.F., Fu, K.S.: A parallel thinning algorithm for 3-D pictures, *Computer Graphics and Image Processing* **17** (1981) 315–331.
- [71] Udupa, J.K., Samarasekera, S.: Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation, *Graphical Models and Image Processing* **58** (1996) 246–261.
- [72] Ulichney, R.: *Digital Halftoning*, The MIT Press, 1987.
- [73] Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns, *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 376–380.
- [74] Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations, *IEEE Trans. Pattern Analysis and Machine Intelligence* **13** (1991) 583–598.
- [75] Viola, P., Wells, W.M., III: Alignment by Maximization of Mutual Information, in *Proc. IEEE International Conference on Computer Vision*, 16–23, 1995.
- [76] Voss, K.: *Discrete Images, Objects, and Functions in  $Z^n$* , Springer-Verlag, 1993.
- [77] Watt, A., Policarpo, F.: *The computer image*, Addison-Wesley, 1998.
- [78] Wells, W.M., Viola, P., Atsumi, H., Nakajima, S., Kikinis, R.: Multi-modal volume registration by maximization of mutual information, *Medical Image Analysis* **1** (1996) 35–51.



- [79] Yamada, H.: Complete Euclidean distance transformation by parallel operation, in *Proc. 7th Int. Conf. Pattern Recognition*, 69–71, 1984.
- [80] Zadeh, L.A.: Fuzzy sets, *Information and Control* **8** (1965) 338–353.